

# Principal minors Pfaffian half-tree theorem

Béatrice de Tilière \*

## Abstract

A half-tree is an edge configuration whose superimposition with a perfect matching is a tree. In this paper, we prove a half-tree theorem for the Pfaffian principal minors of a skew-symmetric matrix whose column sum is zero; introducing an explicit algorithm, we fully characterize half-trees involved. This question naturally arose in the context of statistical mechanics where we aimed at relating perfect matchings and trees on the same graph. As a consequence of the Pfaffian half-tree theorem, we obtain a refined version of the matrix-tree theorem in the case of skew-symmetric matrices, as well as a line-bundle version of this result.

## 1 Introduction

We prove a half-tree theorem for the Pfaffian principal minors of a skew-symmetric matrix whose column sum is zero. This is a Pfaffian version of the classical matrix-tree theorem [Kir47], see also [Cha82] and references therein. Introducing an explicit algorithm, we give a constructive proof of our result and a full characterization of half-trees involved. A precise statement of our main theorem, as well as consequences for the determinant, are given in Section 1.1 of the introduction. An outline of the paper is provided in Section 1.2. Motivations for proving such a result come from statistical mechanics and are exposed in Section 1.3.

### 1.1 Statement of main result

Let  $V^R = V \cup R$ , where  $V = \{1, \dots, n\}$ ,  $R = \{n+1, \dots, n+r\}$  and  $n$  is even. Let  $A^R = (a_{ij})_{\{i,j \in V^R\}}$  be a skew-symmetric whose column sum is zero and let  $A = (a_{ij})_{\{i,j \in V\}}$  be the matrix obtained from  $A^R$  by removing the  $r$  last lines and columns. The matrix  $A$  is also skew-symmetric and the *Pfaffian* of  $A$ , denoted  $\text{Pf}(A)$ , is:

$$\text{Pf}(A) = \sum_{\pi \in \mathcal{P}} \text{sgn}(\sigma_\pi) a_{\sigma_\pi(1)\sigma_\pi(2)} \cdots a_{\sigma_\pi(n-1)\sigma_\pi(n)},$$

where  $\mathcal{P}$  is the set of partitions of  $\{1, \dots, n\}$  into  $n/2$  unordered pairs, also known as *pairings*, and  $\sigma_\pi$  is a permutation such that  $\{\sigma_\pi(1)\sigma_\pi(2), \dots, \sigma_\pi(n-1)\sigma_\pi(n)\}$  is a description of the partition  $\pi$ . If  $n$  is odd, then by convention  $\text{Pf}(A) = 0$ .

---

\*Laboratoire de Probabilités et Modèles Aléatoires, Université Pierre et Marie Curie, 4 place Jussieu, F-75005 Paris. [beatrice.de.tiliere@upmc.fr](mailto:beatrice.de.tiliere@upmc.fr).

To the matrix  $A^R$  one associates the graph  $G^R = (V^R, E^R)$ , where  $E^R = \{ij : i, j \in V^R, a_{ij} \neq 0\}$ . Every oriented edge  $(i, j)$  of  $G^R$  is assigned a weight  $a_{ij}$ , thus defining a skew-symmetric weight function on oriented edges. The matrix  $A^R$  is the *weighted adjacency matrix* of the graph  $G^R$ .

A *spanning forest* of  $G^R$  is an oriented edge configuration of  $G^R$ , spanning vertices of  $V$ , such that each connected component is a tree containing exactly one vertex of  $R$ . This vertex is taken to be the root and edges of the component are oriented towards it. Equivalently, a spanning forest of  $G^R$  is an oriented edge configuration containing no cycles, such that each vertex of  $V$  has exactly one outgoing edge of the configuration. A *leaf* of a spanning forest is a vertex with no incoming edge.

Let  $G = (V, E)$  be the graph naturally associated to the matrix  $A$ . A *perfect matching*  $M_0$  of  $G$  is a subset of edges such that each vertex of  $V$  is incident to exactly one edge of  $M_0$ . Note that a perfect matching of  $G$  contains exactly  $|V|/2$  edges. In the whole of this paper, we suppose that  $G$  has at least one perfect matching; if this is not the case, then  $\text{Pf}(A) = 0$  (see also Section 2.1). We let  $\mathcal{M}$  denote the set of perfect matchings.

Let  $F$  be a spanning forest of  $G^R$ , then  $F$  is said to be *compatible* with  $M_0$  if it consists of the  $|V|/2$  edges of  $M_0$  and of  $|V|/2$  edges of  $E^R \setminus M_0$ . The oriented edge configuration  $F \setminus M_0$  is referred to as a *half-spanning forest*. In the specific case where  $R$  is reduced to a point, then  $F$  is a tree and  $F \setminus M_0$  is referred to as a *half-tree*.

EXAMPLE. Let  $V^R = \{1, 2, 3, 4, 5\}$ ,  $V = \{1, 2, 3, 4\}$ ,  $R = \{5\}$ . Consider the graphs  $G^R$  and  $G$  pictured in Figure 1 below. A choice of perfect matching  $M_0$  of  $G$  is pictured in white, and  $F_1, F_2, F_3$  are examples of spanning trees of  $G^R$  compatible with  $M_0$ .

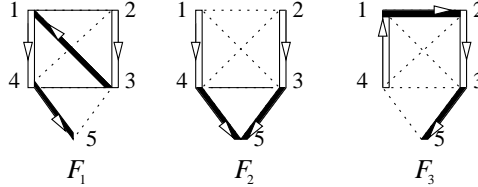


Figure 1: Spanning trees compatible with  $M_0$ .

Here is the statement of our main theorem.

**Theorem 1** (Principal minors Pfaffian half-tree theorem). *Let  $A^R$  be a skew-symmetric matrix of size  $(n+r) \times (n+r)$ , whose column sum is zero, such that  $n$  is even; and let  $A$  be the matrix obtained from  $A^R$  by removing the  $r$  last lines and columns. Let  $G^R$  and  $G$  be the graphs naturally constructed from the matrices  $A^R$  and  $A$ , respectively.*

*For every perfect matching  $M_0$  of  $G$ , the Pfaffian of  $A$  is equal to:*

$$\text{Pf}(A) = \sum_{F \in \mathcal{F}(M_0)} \text{sgn}(\sigma_{M_0(F \setminus M_0)}) \prod_{e \in F \setminus M_0} a_e,$$

*where  $a_e$  is the coefficient of the matrix  $A^R$  corresponding to the oriented edge  $e$ ;  $\text{sgn}(\sigma_{M_0(F \setminus M_0)})$  is the signature of the permutation  $\sigma_{M_0(F \setminus M_0)}$  of Definition 1.1 below;  $\mathcal{F}(M_0)$  is the set*

of spanning forests of  $G^R$  compatible with  $M_0$ , satisfying Condition (C) of Definition 1.2 below.

**Definition 1.1.** Let  $F$  be a spanning forest of  $G^R$  compatible with  $M_0$ . The orientation of  $F$  induces an orientation of edges of the perfect matching  $M_0$ , and we let  $(i_1, i_2), \dots, (i_{n-1}, i_n)$  be a description of the oriented matching. Then,  $\sigma_{M_0(F \setminus M_0)}$  is the permutation:

$$\sigma_{M_0(F \setminus M_0)} = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

Note that the interchange of two pairs does not change the signature of the permutation.

Here is the algorithm used to characterize half-spanning forests which contribute to  $\text{Pf}(A)$ .

**Trimming algorithm**

Input: a spanning forest  $F$  of  $G^R$  compatible with  $M_0$ .

Initialization:  $F_1 = F$ .

Step  $i, i \geq 1$

Let  $\ell_1^i$  be the largest leaf of  $F_i$  and consider the connected component containing  $\ell_1^i$ . Start from  $\ell_1^i$  along the unique path joining  $\ell_1^i$  to the root of this component, until the first time one of the following vertices is reached:

- the root vertex,
- a fork, that is a vertex with more than one incoming edge,
- a vertex which is smaller than the leaf  $\ell_1^i$ .

This yields a loopless path  $\lambda_{\ell_1^i}$  starting from  $\ell_1^i$ , of length  $\geq 1$ . Let  $F_{i+1} = F_i \setminus \lambda_{\ell_1^i}$ . If  $F_{i+1}$  is empty, stop; else go to Step  $i + 1$ .

End: since edges are removed at every step, and since  $F$  contains a finite number of edges, the trimming algorithm ends in finite time  $N$ .

**Definition 1.2.** A spanning forest  $F$  compatible with  $M_0$  is said to satisfy *Condition (C)* if each of the paths  $\lambda_{\ell_1^1}, \dots, \lambda_{\ell_1^N}$  obtained from the trimming algorithm, starts from an edge of  $M_0$  and has even length. We let  $\mathcal{F}(M_0)$  be the set of spanning forests compatible with  $M_0$ , satisfying Condition (C).

EXAMPLE. Applying the trimming algorithm to each of the spanning forests  $F_1, F_2, F_3$  of Figure 1 yields:

$$\begin{aligned} F_1 : & \text{ Step 1: } \ell_1^1 = 2, \lambda_2 = 2, 3, 1. \quad \text{Step 2: } \ell_1^2 = 1, \lambda_1 = 1, 4, 5. \\ F_2 : & \text{ Step 1: } \ell_1^1 = 2, \lambda_2 = 2, 3, 5. \quad \text{Step 2: } \ell_1^2 = 1, \lambda_1 = 1, 4, 5. \\ F_3 : & \text{ Step 1: } \ell_1^1 = 4, \lambda_4 = 4, 1. \quad \text{Step 2: } \ell_1^2 = 1, \lambda_1 = 1, 2, 3, 5. \end{aligned}$$

The spanning trees  $F_1$  and  $F_2$  satisfy Condition (C) but not  $F_3$ .

*Remark 2.*

- It is interesting to note that taking different perfect matchings  $M_0$  yields different families of half-spanning forests. It is not clear a priori, without using the Pfaffian half-tree theorem, that these families should have the same total weight.
- Suppose that we change the labeling of the vertices. Let  $\tilde{A}^R$  be the corresponding skew-symmetric adjacency matrix, and  $\tilde{A}$  be the matrix obtained by removing the  $r$  last lines and columns. As long as the re-labeling does not affect vertices of  $R$ , the matrix  $\tilde{A}$  is obtained from the matrix  $A$  by exchanging lines and columns, so that  $\text{Pf}(\tilde{A}) = \pm \text{Pf}(A)$ . Applying the Pfaffian half-tree theorem to the matrices  $A$  and  $\tilde{A}$  nevertheless yields a different set of half-spanning forests and again, it is not clear a priori that they should have the same total weight in absolute value. Note that taking other principal minors amounts to changing the labeling of the vertices.
- In the paper [MV02], Masbaum and Vaintrob assign to a 3-uniform hypergraph a specific skew-symmetric matrix whose column sum is zero, and prove that the Pfaffian of any principal minor of this matrix enumerates signed spanning trees of the 3-uniform graph. A combinatorial proof of their result is then given by Hirschman and Reiner [HR04], and yet another proof using Grassman variables is provided by Abdesselam [Abd04]. The matrix considered by Masbaum and Vaintrob satisfies the assumptions of Theorem 1, and it would be interesting to investigate whether there is an explicit relation between half-spanning forests of Theorem 1 and spanning trees of the 3-uniform graph.

Using the fact that the determinant of a skew-symmetric matrix is the square of the Pfaffian, we obtain the following corollary.

**Corollary 3.** *Let  $A^R$  be a skew-symmetric matrix of size  $(n+r) \times (n+r)$ , whose column sum is zero, such that  $n$  is even; and let  $A$  be the matrix obtained from  $A^R$  by removing the  $r$  last lines and columns. Let  $G^R$  and  $G$  be the graphs naturally constructed from the matrices  $A^R$  and  $A$  respectively.*

*The determinant of the matrix  $A$  is equal to:*

$$\det(A) = \sum_{M_0 \in \mathcal{M}} \sum_{F \in \mathcal{F}(M_0)} \prod_{e \in F} a_e,$$

*where  $a_e$  is the coefficient of the matrix  $A^R$  corresponding to the oriented edge  $e$ , and  $\mathcal{F}(M_0)$  is the set of spanning forests compatible with  $M_0$ , satisfying Condition (C).*

*Remark 4.*

- The fact that principal minors of a skew-symmetric matrix whose column sum is zero, count spanning forests is also a consequence of the more general all-minors matrix-tree theorem (which holds for any matrix whose column sum is zero). A combinatorial way of proving this result is to use the explicit expansion of configurations due to Chaiken [Cha82]. This method is not satisfactory in our context, since it does not shed a light on how spanning forests are obtained from double perfect

matchings, which is what we aim for, see Section 1.3. Indeed, the idea of Chaiken’s proof is to expand terms on the diagonal of the matrix and show that only spanning forests remain. In the case of skew-symmetric matrices, since diagonal terms are 0 this amounts to ‘artificially’ creating configurations which do not exist. As a result of our proof, we explicitly construct spanning forests from double perfect matchings, and identify a specific family of spanning forests counted by principal minors. In particular, this implies that in the case of skew-symmetric matrices, specific cancellations occur within spanning forests of the general matrix-tree theorem, a fact hard to establish without using Corollary 3.

- An intrinsic definition of  $\cup_{M_0 \in \mathcal{M}} \mathcal{F}(M_0)$ , not using reference perfect matchings, is given in Remark 16 of Section 3.2.
- A line bundle version of this result, in the spirit of [For93] and [Ken11], is proved in Section 3.3, see Corollary 19.

## 1.2 Outline of the paper

- Section 2. In Section 2.1, we state the interpretation of the Pfaffian as counting signed perfect matchings of the graph  $G$ . Fixing a reference perfect matching  $M_0$ , we then introduce an explicit algorithm, which constructs from the superimposition of  $M_0$  and a generic perfect matching  $M$  counted by the Pfaffian, a family of half RC-spanning forests whose connected components are trees rooted on vertices of  $R$ , or on cycles of even length  $\geq 4$ ; and whose total weight is equal to the contribution of  $M$  to the Pfaffian. The main tool of the algorithm is the ‘opening’ of doubled edges procedure, described in Section 2.3. Step 1 of the algorithm is exposed in Section 2.4, and the full algorithm is the subject of Section 2.5. A characterization of configurations obtained is given in Section 2.6.
- Section 3. Section 3.1 consists in the proof of Theorem 1. The idea is to show that the contribution of half RC-spanning forests constructed above, having connected components rooted on cycles of length  $\geq 4$  cancel, and that only the contribution of spanning forests (rooted on vertices of  $R$ ) remains. The characterization of configurations obtained from the algorithm is also simplified in the case of spanning forests, yielding the trimming algorithm of Section 1.1 of the introduction. The proof of Corollary 3 is the subject of Section 3.2. Finally, in Section 3.3, Corollary 19 proves a line bundle version of the matrix-tree theorem for skew-symmetric matrices of Corollary 3.

## 1.3 A question from statistical mechanics

The Pfaffian half-tree theorem answers a question raised when working on the paper [dT], in which we prove an explicit relation, on the level of configurations, between two models of statistical mechanics: the dimer model on the Fisher graph corresponding to the low temperature expansion of the critical Ising model (through Fisher’s correspondence [Fis66]), and spanning forests. The question raised does not rely on the Fisher graph and can be rephrased in the following, more general framework.

In the setting of statistical mechanics, a perfect matching of a graph is known as a *dimer configuration*. Assigning non-negative weights to edges of the graph naturally defines a weight for each dimer configuration (by taking the product of the edge-weights present in the configuration) and a probability measure on all dimer configurations of  $G$ , thus yielding a statistical mechanics model. The dimer model on planar graphs has been the subject of extensive studies in the last 50 years, and of huge progresses in the last 15 years, see [Ken09] for an overview.

A *double dimer configuration* is the superimposition of two dimer configurations. Our goal is to explicitly construct spanning forests from double dimer configurations when the model is critical, and to do so on the same graph, thus relating two models of statistical mechanics on the level of configurations.

When the graph is planar, dimer configurations are counted by the Pfaffian of the *Kasteleyn matrix* [Kas67, TF61], which is a skew-symmetric matrix with a specific choice of orientation of the edges compensating signs of signatures. The square of the Pfaffian, which is the determinant, counts double dimer configurations. In the case of the dimer model corresponding to the critical Ising model the column sum is zero (when the graph is embedded on the torus), a fact related to the model being critical. This motivates our choice of taking the column sum equal to zero. In general, the Pfaffian of a skew-symmetric matrix counts signed perfect matchings, with the sign coming from the signature of a permutation naturally assigned to the matching, see Section 2.1. Theorem 1 proves that principal minors of the Pfaffian of a general skew-symmetric whose column sum is zero counts a specific family of half-spanning forests, and half-spanning forests are explicitly constructed from perfect matchings. Corollary 3 proves that principal minors of the determinant of such a matrix counts a family of spanning forests, and the latter are explicitly constructed from double perfect matchings. Specifying this result to the case of planar graphs or graphs embedded on the torus, and Kasteleyn matrices, answers our initial question.

It is known from the work of Temperley [Tem72], Kenyon, Propp and Wilson [KPW00] that spanning trees are in bijection with dimer configurations of a related bipartite graph. Conversely, Kenyon and Sheffield [KS04] relate dimer configurations of a bipartite graph to spanning trees of a process on an associated graph. In this paper perfect matchings and trees live on the same graph, the graph must not be bipartite nor even planar, the weight function on edges of the graph must not be positive, but the column sum must be zero. The main drawback of our result in the context of statistical mechanics is that, even when the matrix is Kasteleyn and perfect matchings all have positive weights, corresponding spanning forests might have negative weights.

## 2 From matchings to half $RC$ -rooted spanning forests

Let us recall the setting:  $A^R$  is a skew-symmetric matrix of size  $(n+r) \times (n+r)$ , whose column sum is zero, such that  $n$  is even; and  $A$  is the matrix obtained from  $A^R$  by removing the  $r$  last lines and columns;  $G^R$  and  $G$  are the graphs naturally constructed from the matrices  $A^R$  and  $A$  in Section 1.1 of the introduction.

**Definition 2.1.** An *RC-rooted spanning forest*, referred to as an RCRSF is an oriented edge configuration of  $G^R$  spanning vertices of  $G$ , such that each connected component is, either a tree rooted on a vertex of  $R$ , or a tree rooted on a cycle of  $G$ , which we refer to as a *unicycle*. Edges of each of the components are oriented towards its root, and edges of the cycles are oriented in one of the two possible directions.

**Definition 2.2.** Let  $M_0$  be a reference perfect matching of  $G$ . An RCRSF  $F$  is said to be *compatible with  $M_0$* , if it consists of the  $|V|/2$  edges of  $M_0$ , and of  $|V|/2$  edges of  $E^R \setminus M_0$ . Moreover cycles of uni-cycles have even length  $\geq 4$ , and alternate between edges of  $M_0$  and  $F \setminus M_0$ . The oriented edge configuration  $F \setminus M_0$  is referred to as a *half-RCRSF*.

In Section 2.1, we give the graphical interpretation of the Pfaffian of the matrix  $A$  as counting signed perfect matchings of  $G$ . Let  $M$  be a generic perfect matching counted by the Pfaffian and  $M_0$  be a fixed reference perfect matching of  $G$ . In Sections 2.4 and 2.5, we introduce an explicit algorithm which constructs, from the superimposition of  $M_0$  and  $M$ , a family of half *RC*-rooted spanning forests compatible with  $M_0$ , whose total weight is equal to the contribution of  $M$  to the Pfaffian. In Section 2.6, we characterize *RC*-spanning forests obtained. Notations used are given in Section 2.2. The main graphical idea of the algorithm is the subject of Section 2.3.

## 2.1 Graphical interpretation of the Pfaffian

Recall that  $\mathcal{P}$  denotes the set of pairings of  $\{1, \dots, n\}$ , and let  $\mathcal{M}$  be the set of perfect matchings of  $G$ . Observing that every perfect matching of  $G$  corresponds to a pairing of  $\mathcal{P}$ , and that pairings of  $\mathcal{P}$  which do not correspond to perfect matchings of  $G$  contribute 0 to the Pfaffian, we can rewrite  $\text{Pf}(A)$  as:

$$\text{Pf}(A) = \sum_{M \in \mathcal{M}} \text{sgn}(\sigma_M) a_{\sigma_M(1)\sigma_M(2)} \cdots a_{\sigma_M(n-1)\sigma_M(n)},$$

where  $\sigma_M$  is a permutation such that  $\{\sigma_M(1)\sigma_M(2), \dots, \sigma_M(n-1)\sigma_M(n)\}$  is a description of the perfect matching  $M$ .

Choosing the permutation  $\sigma_M$  amounts to choosing an order for the  $n/2$  pairs and an order for the two elements of each of the pairs, meaning that there are  $(\frac{n}{2})! 2^{\frac{n}{2}}$  choices. Exchanging two pairs does not change the signature nor the corresponding coefficients of the matrix, whereas changing two elements of a pair changes the signature of the permutation and the sign of the corresponding element of the matrix. As a consequence, the global sign is unchanged, and fixing the signature amounts to choosing an orientation of edges of the perfect matching.

We now specify the choice of signature of the permutation  $\sigma_M$  by choosing an orientation of edges of  $M$ . Let  $M_0$  be a fixed reference matching of  $G$ . The superimposition of  $M_0$  and  $M$ , denoted by  $M_0 \cup M$ , consists of disjoint doubled edges (covered by both  $M_0$  and  $M$ ) and alternating cycles of even length  $\geq 4$ , covering all vertices of  $G$ . Let us, for the moment, consider doubled edges as cycles of length 2. The orientation of the superimposition  $M_0 \cup M$  is fixed by the following rule: for each cycle, the orientation is compatible with that of the edge  $(\ell_1, \ell'_1)$ , where  $\ell_1$  is the smallest vertex of the cycle, and

$\ell'_1$  is its partner in  $M_0$ . This yields an orientation of edges of  $M$ , and thus a choice of  $\sigma_M$ . This procedure also gives an orientation of edges of  $M_0$ , and thus a choice of  $\sigma_{M_0}$ . Let us also denote by  $M_0 \cup M$  the oriented superimposition.

EXAMPLE (FIGURE 2). In white is a choice of reference perfect matching  $M_0$  and in black are the three possible matchings  $M_1, M_2, M_3$  of the graph  $G$ . Edges of the respective superimpositions are oriented according to the rule described above.

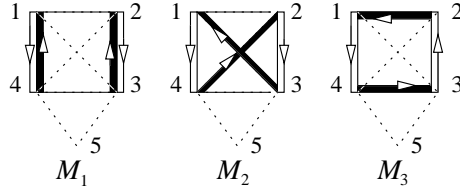


Figure 2: Oriented superimposition.

Let  $\sigma_{M_0 \cup M}$  be the permutation whose cyclic decomposition corresponds to cycles of the superimposition  $M_0 \cup M$ . Then,

$$\text{sgn}(\sigma_{M_0 \cup M}) = (-1)^{|\mathcal{D}(M_0 \cup M)|} (-1)^{|\mathcal{C}(M_0 \cup M)|},$$

where  $\mathcal{D}(M_0 \cup M)$  is the set of doubled edges of  $M_0 \cup M$  and  $\mathcal{C}(M_0 \cup M)$  is the set of alternating cycles of length  $\geq 4$  of  $M_0 \cup M$ . Note that this signature does not depend on the orientation of the cycles. Following Kasteleyn [Kas67], signatures of the permutations  $\sigma_{M_0}$  and  $\sigma_M$  are related as follows:

$$\begin{aligned} \text{sgn}(\sigma_M) &= \text{sgn}(\sigma_{M_0}) \text{sgn}(\sigma_{M_0 \cup M}) \\ &= \text{sgn}(\sigma_{M_0}) (-1)^{|\mathcal{D}(M_0 \cup M)|} (-1)^{|\mathcal{C}(M_0 \cup M)|}. \end{aligned}$$

Writing  $\sigma_{M_0}$  as  $\sigma_{M_0(M)}$  to remember that our choice of orientation of edges of  $M_0$  depends on  $M$ , the Pfaffian of  $A$  can be expressed as:

$$\text{Pf}(A) = \sum_{M \in \mathcal{M}} w_{M_0}(M), \quad (1)$$

where

$$w_{M_0}(M) = \text{sgn}(\sigma_{M_0(M)}) (-1)^{|\mathcal{D}(M_0 \cup M)|} (-1)^{|\mathcal{C}(M_0 \cup M)|} \prod_{e \in M} a_e,$$

and  $a_e$  is the coefficient of the matrix  $A$  corresponding to the oriented edge  $e$  of  $M$ .

## 2.2 Notations

Let  $M_0$  be a fixed reference perfect matching of the graph  $G$ ,  $M$  be a generic perfect matching, and  $M_0 \cup M$  be the oriented superimposition of  $M_0$  and  $M$  constructed in Section 2.1. In order to shorten notations, we write  $\mathcal{D}$  instead of  $\mathcal{D}(M_0 \cup M)$  for the set



of doubled edges of the superimposition,  $\mathcal{C}$  instead of  $\mathcal{C}(M_0 \cup M)$  for the set of cycles of length  $\geq 4$ , and  $w(M)$  instead of  $w_{M_0}(M)$ .

We now introduce definitions and notations used in the algorithm of Sections 2.4 and 2.5. Let  $V^{\mathcal{C}}$  denote the set of vertices of  $V$  which belong to a cycle of  $\mathcal{C}$ . For every subset  $\mathcal{D}'$  of doubled edges of  $\mathcal{D}$ , let  $V^{\mathcal{D}'}$  denote the set of vertices of  $V$  which belong to doubled edges of  $\mathcal{D}'$ .

Every vertex  $i \in V^{\mathcal{D}}$  belongs to a doubled edge covering vertices  $i$  and  $i'$  of  $\mathcal{D}$ . We denote by  $\mathbf{e}_i$  (or  $\mathbf{e}'_i$ ) this doubled edge and define  $V_i$  to be the set of vertices in the full graph  $G^R$ , adjacent to  $i'$  other than  $i$ .

For every subset  $\mathcal{D}'$  of  $\mathcal{D}$ , denote by  $V_i^{\mathcal{D}'}$  the set of vertices of  $V_i$  which belong to doubled edges of  $\mathcal{D}'$ , and by  $(V_i^{\mathcal{D}'})^c$  those which don't. Then  $V_i$  can be partitioned as:  $V_i = V_i^{\mathcal{D}'} \cup (V_i^{\mathcal{D}'})^c$ .

### 2.3 Idea of the algorithm

The idea of the algorithm is to use the reference configuration  $M_0$  as a skeleton for opening up doubled edges of the superimposition  $M_0 \cup M$ . Indeed, because of the condition  $\sum_{j=1}^n a_{i,j} = 0$ , configurations of Figure 3 have opposite weights.

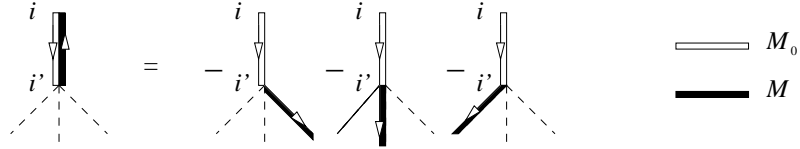


Figure 3: ‘Opening’ of doubled edges procedure:  $a_{i',i} = - \sum_{j \in V_i} a_{i',j}$ .

There are two main difficulties in realizing this procedure: the first is that there is, a priori, no natural way of deciding whether to ‘open’ up of the doubled edge at the vertex  $i$  or at the vertex  $i'$ . The second is that we want to keep track of configurations constructed, show that we obtain  $RC$ -rooted spanning forests, characterize them and prove that only spanning forests remain. It turns out that the ‘opening’ procedure depends strongly on the labeling of vertices.

### 2.4 Algorithm: Step 1

Recall that the goal of the algorithm is to construct, from the superimposition  $M_0 \cup M$  of a reference perfect matching  $M_0$  and a generic perfect matching  $M$ , a family of half- $RC$ -rooted spanning forests of  $G^R$  compatible with  $M_0$ , whose total weight is equal to the contribution of  $M$  to the Pfaffian. In this section, we introduce the first step of the algorithm, setting rules for the opening up of doubled edges of  $M_0 \cup M$ . The full algorithm, which in essence consists of iterations of Step 1, is the subject of Section 2.5.

Input: oriented superimposition  $M_0 \cup M$ .

Initialization: if the superimposition  $M_0 \cup M$  consists of cycles only, that is if the set  $\mathcal{D}$  is empty, let  $\mathcal{O}_0 = \{M\}$  and stop. Else, let  $\mathcal{O}_0 = \{\emptyset\}$  and go to the first iteration.

EXAMPLE (FIGURE 4). Consider Figure 4 as input of the algorithm. The algorithm will be explicitly performed on this example, throughout Sections 2.4 and 2.5.

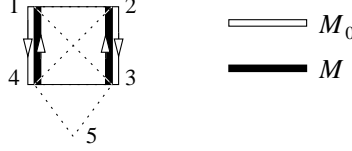


Figure 4: Input  $M_0 \cup M$  of Step 1.

Since  $M_0 \cup M$  contains doubled edges, the output  $\mathcal{O}_0$  is  $\{\emptyset\}$ .

#### Iteration 1

- Define  $\ell_1 = \min\{i \in V : i \text{ belongs to a doubled edge of } \mathcal{D}\}$ . Then  $\ell_1$  is the partner of a vertex  $\ell'_1$  in  $M_0$  and  $M$ . By our choice of orientation for  $M_0$  and  $M$ , the edge  $\ell_1 \ell'_1$  is oriented from  $\ell_1$  to  $\ell'_1$  in  $M_0$  and from  $\ell'_1$  to  $\ell_1$  in  $M$ . For every  $\ell_2 \in V_{\ell_1}$ , define:

$$M_{\ell_1, \ell_2} = \{M \setminus (\ell'_1, \ell_1)\} \cup \{(\ell'_1, \ell_2)\}$$

$$w(M_{\ell_1, \ell_2}) = \text{sgn}(\sigma_{M_0(M)}) (-1)^{|\mathcal{D}|-1} (-1)^{|\mathcal{C}|} \prod_{e \in M_{\ell_1, \ell_2}} a_e.$$

EXAMPLE (FIGURE 5):  $\ell_1 = 1$ ,  $V_{\ell_1} = V_1 = \{2, 3, 5\}$ , yielding configurations  $M_{1,2}$ ,  $M_{1,3}$ ,  $M_{1,5}$ .

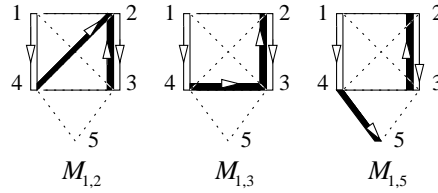


Figure 5: From left to right: black edges are the oriented edge configurations  $M_{1,2}$ ,  $M_{1,3}$ ,  $M_{1,5}$ .

- Let  $D_{\ell_1}$  be the set of doubled edges  $\mathcal{D} \setminus \{e_{\ell_1}\}$ . Then, the set  $V_{\ell_1}$  can be partitioned as the set of vertices of  $V_{\ell_1}$  which belong to a doubled edge of  $\mathcal{D}_{\ell_1}$  and the set of those which don't. Using notations of Section 2.2, this can be rewritten as:  $V_{\ell_1} = V_{\ell_1}^{\mathcal{D}_{\ell_1}} \cup (V_{\ell_1}^{\mathcal{D}_{\ell_1}})^c$ .

The output of Iteration 1 is the set of configurations  $M_{\ell_1, \ell_2}$  such that  $\ell_2$  does not belong to a doubled edge of  $\mathcal{D}_{\ell_1}$ :

$$\begin{aligned}\mathcal{O}_1 &= \bigcup_{\ell_2 \in (V_{\ell_1}^{\mathcal{D}_{\ell_1}})^c} M_{\ell_1, \ell_2}, \\ w(\mathcal{O}_1) &= \sum_{M_{\ell_1, \ell_2} \in \mathcal{O}_1} w(M_{\ell_1, \ell_2}).\end{aligned}$$

where by convention, if  $(V_{\ell_1}^{\mathcal{D}_{\ell_1}})^c = \emptyset$ , then  $\mathcal{O}_1 = \emptyset$  and  $w(\mathcal{O}_1) = 0$ .

- The algorithm continues with configurations  $M_{\ell_1, \ell_2}$  where  $\ell_2$  belongs to a doubled edge of  $\mathcal{D}_{\ell_1}$ . Formally we have: if  $V_{\ell_1}^{\mathcal{D}_{\ell_1}} = \emptyset$ , then stop; else, go to Iteration 2.

EXAMPLE: the set  $\mathcal{D}_{\ell_1} = \mathcal{D}_1$  consists of the doubled edge 23. As a consequence, the set  $V_{\ell_1} = V_1 = \{2, 3, 5\}$  is partitioned as  $V_1 = \{2, 3\} \cup \{5\}$ , and the output of iteration 1 is  $\mathcal{O}_1 = \{M_{1,5}\}$ . The algorithm continues with  $M_{1,2}$  and  $M_{1,3}$ .

#### Iteration $k$ , ( $k \geq 2$ )

For every  $\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}, \dots, \ell_k \in V_{\ell_{k-1}}^{\mathcal{D}_{\ell_1, \dots, \ell_{k-1}}}$ , do the following.

- The vertex  $\ell_k$  is the partner of a vertex  $\ell'_k$  in  $M_0$  and  $M$  (since  $\mathcal{D}_{\ell_1, \dots, \ell_{k-1}}$  is a subset of  $\mathcal{D}$ ). If  $\ell_k < \ell'_k$ , then by our choice of orientation, the edge  $\ell_k \ell'_k$  is oriented from  $\ell_k$  to  $\ell'_k$  in  $M_0$  and from  $\ell'_k$  to  $\ell_k$  in  $M_{\ell_1, \dots, \ell_k}$ . If  $\ell_k > \ell'_k$ , then we change the orientation of this edge in  $M_0$  and in  $M_{\ell_1, \dots, \ell_k}$ . Let us also denote by  $M_{\ell_1, \dots, \ell_k}$  this new configuration. This change of orientation has the effect of changing the permutation assigned to  $M_0$ , and we denote by  $\sigma_{M_0(M_{\ell_1, \dots, \ell_k})}$  this new permutation. It also negates the contribution of  $M_{\ell_1, \dots, \ell_k}$  so that the global contribution remains unchanged. For every  $\ell_{k+1} \in V_{\ell_k}$ , define:

$$\begin{aligned}M_{\ell_1, \dots, \ell_{k+1}} &= (M_{\ell_1, \dots, \ell_k} \setminus (\ell'_k, \ell_k)) \cup (\ell'_k, \ell_{k+1}) \\ w(M_{\ell_1, \dots, \ell_{k+1}}) &= \text{sgn}(\sigma_{M_0(M_{\ell_1, \dots, \ell_k})}) (-1)^{|\mathcal{D}| - k} (-1)^{|\mathcal{C}|} \prod_{e \in M_{\ell_1, \dots, \ell_{k+1}}} a_e\end{aligned}\quad (2)$$

EXAMPLE (FIGURE 6). Recall that  $V_{\ell_1}^{\mathcal{D}_{\ell_1}} = V_1^{\{23\}} = \{2, 3\}$ , so that  $\ell_2 \in \{2, 3\}$ . If  $\ell_2 = 2$ , then  $V_{\ell_2} = V_2 = \{1, 4, 5\}$ , yielding configurations  $M_{1,2,1}$ ,  $M_{1,2,4}$ ,  $M_{1,2,5}$ . If  $\ell_2 = 3$ , then  $V_{\ell_2} = V_3 = \{1, 4\}$ , yielding configurations  $M_{1,3,1}$ ,  $M_{1,3,4}$ .

- Let  $\mathcal{D}_{\ell_1, \dots, \ell_k}$  be the set of doubled edges  $\mathcal{D}_{\ell_1, \dots, \ell_{k-1}} \setminus \{\mathbf{e}_{\ell_k}\}$ . Then, the set  $V_{\ell_k}$  can be partitioned as:  $V_{\ell_k} = V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}} \cup (V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}})^c$ , and the output of Iteration  $k$  is the set of configurations  $M_{\ell_1, \dots, \ell_{k+1}}$  such that  $\ell_{k+1}$  does not belong to a doubled edge of  $\mathcal{D}_{\ell_1, \dots, \ell_k}$ .

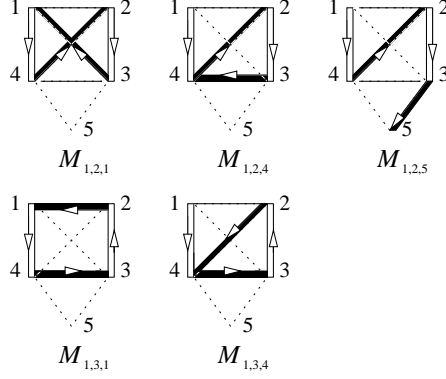


Figure 6: First line, from left to right, black edges consists of the configurations  $M_{1,2,1}$ ,  $M_{1,2,4}$ ,  $M_{1,2,5}$ . Second line, from left to right, black edges consists of the configurations  $M_{1,3,1}$ ,  $M_{1,3,4}$ .

$$\mathcal{O}_k = \bigcup_{\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}} \dots \bigcup_{\ell_k \in V_{\ell_{k-1}}^{\mathcal{D}_{\ell_1, \dots, \ell_{k-1}}}} \bigcup_{\ell_{k+1} \in (V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}})^c} M_{\ell_1, \dots, \ell_{k+1}},$$

$$w(\mathcal{O}_k) = \sum_{M_{\ell_1, \dots, \ell_{k+1}} \in \mathcal{O}_k} w(M_{\ell_1, \dots, \ell_{k+1}}).$$

- If  $V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}} = \emptyset$ , then stop. Else, go to Step  $k + 1$ .

EXAMPLE: when  $\ell_2 = 2$ , the set  $\mathcal{D}_{1,2}$  is empty so that  $V_2$  is partitioned as  $\{\emptyset\} \cup \{1, 4, 5\}$  and the contribution to the output  $\mathcal{O}_2$  of iteration 2 is  $M_{1,2,1}$ ,  $M_{1,2,4}$ ,  $M_{1,2,5}$ . When  $\ell_2 = 3$ , the set  $\mathcal{D}_{1,3}$  is also empty, implying that  $V_3$  is partitioned as  $\{\emptyset\} \cup \{1, 4\}$  and the contribution to the output  $\mathcal{O}_2$  of iteration 2 is  $M_{1,3,1}$ ,  $M_{1,3,4}$ . After Iteration 2, for every  $\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}$ , the set  $V_{\ell_2}^{\mathcal{D}_{\ell_1, \ell_2}}$  is empty, so that the algorithm stops.

End

Step 1 of the algorithm stops at time  $m$  for the first time, if it hasn't stopped at time  $m - 1$ , and if for every  $\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}, \dots, \ell_m \in V_{\ell_{m-1}}^{\mathcal{D}_{\ell_1, \dots, \ell_{m-1}}}; V_{\ell_m}^{\mathcal{D}_{\ell_1, \dots, \ell_m}} = \emptyset$ . This implies in particular that  $(V_{\ell_m}^{\mathcal{D}_{\ell_1, \dots, \ell_m}})^c = V_{\ell_m}$ . Since the number of doubled edges decreases by 1 every time an iteration of the algorithm occurs, and since the number of doubled edges in  $\mathcal{D}$  is finite, we are sure that Step 1 of the algorithm stops in finite time.

#### 2.4.1 Output of Step 1, geometric properties of configurations

The output of Step 1 of the algorithm is the set of configurations  $\mathcal{S}_1 = \bigcup_{k=0}^m \mathcal{O}_k$ . The weight of this set is defined to be  $w(\mathcal{S}_1) = \sum_{k=0}^m w(\mathcal{O}_k)$ .

If the initial superimposition  $M_0 \cup M$  consists of cycles only, *i.e.* if the set  $\mathcal{D}$  is empty, then  $m = 0$  and  $\mathcal{S}_1 = \{M\}$ . In all other cases, the set  $\mathcal{S}_1$  can be rewritten in a more compact way as:

$$\mathcal{S}_1 = \bigcup_{\gamma_{\ell_1} \in \Gamma_{\ell_1}} M_{\gamma_{\ell_1}},$$

where:

$$\begin{aligned} \Gamma_{\ell_1} = & \left\{ \gamma_{\ell_1} : \gamma_{\ell_1} \text{ is a path of length } 2k \text{ for some } k \in \{1, \dots, m\} : \gamma_{\ell_1} = \ell_1, \ell'_1, \dots, \ell_k, \ell'_k, \ell_{k+1}, \right. \\ & \ell_1 = \min\{i \in V : i \text{ belongs to a doubled edge of } \mathcal{D}\}, \\ & \forall j \in \{2, \dots, k\}, \ell_j \in V_{\ell_{j-1}}^{\mathcal{D}_{\ell_1, \dots, \ell_{j-1}}} \text{ and } \ell'_j \text{ is the partner of } \ell_j \text{ in } M_0 \text{ and } M, \\ & \left. \ell_{k+1} \in (V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}})^c \right\}. \end{aligned}$$

$$M_{\gamma_{\ell_1}} = M_{\ell_1, \dots, \ell_{k+1}}.$$

Let  $\gamma_{\ell_1} = \ell_1, \ell'_1, \dots, \ell_k, \ell'_k, \ell_{k+1}$  be a generic path of  $\Gamma_{\ell_1}$  for some  $k \in \{1, \dots, m\}$ , and let  $F_{\gamma_{\ell_1}}$  denote the superimposition  $M_0 \cup M_{\gamma_{\ell_1}}$ . The configuration  $F_{\gamma_{\ell_1}}$  and the path  $\gamma_{\ell_1}$  satisfy the following properties.

- The oriented edge configuration  $F_{\gamma_{\ell_1}}$ :
  - (I) has one outgoing edge at every vertex of  $V$ , and contains the path  $\gamma_{\ell_1}$ .
  - (II) has  $k$  doubled edges less than  $M_0 \cup M$ .
- The oriented path  $\gamma_{\ell_1}$ :
  - (III) has even length  $2k$ , is alternating (meaning that edges alternate between  $M_0$  and  $M_{\gamma_{\ell_1}}$ ). It starts from the vertex  $\ell_1$  followed by an edge of  $M_0$ .
  - (IV) The vertex  $\ell_1$  is the smallest vertex belonging to a doubled edge of  $\mathcal{D}$ . The  $2k$  first vertices of  $\gamma_{\ell_1}$  are all distinct and the last vertex  $\ell_{k+1}$  belongs to  $(V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}})^c$ . Observing that:

$$(V_{\ell_k}^{\mathcal{D}_{\ell_1, \dots, \ell_k}})^c = V_{\ell_k} \cap (R \cup V^C \cup \{\ell_1, \ell'_1, \dots, \ell_k, \ell'_k\}),$$

we deduce that one of the following holds.

- (IV)(1) If  $\ell_{k+1} \in R$ , then  $\gamma_{\ell_1}$  is a loopless oriented path from  $\ell_1$  to one of the root vertices of  $R$ , and  $\ell_1$  is a leaf of  $F_{\gamma_{\ell_1}}$ . Since  $R = \{n+1, \dots, n+r\}$ ,  $\ell_1$  is smaller than all vertices of  $\gamma_{\ell_1}$ .
- (IV)(2) If  $\ell_{k+1} \in V^C$ , then  $\gamma_{\ell_1}$  is a loopless oriented path ending at a vertex of one of the cycles of  $\mathcal{C}$  that is, the connected component containing  $\ell_1$  is a unicycle with a unique branch. The vertex  $\ell_1$  is a leaf of  $F_{\gamma_{\ell_1}}$  and is smaller than the  $2k$  first vertices of the path, but cannot be a priori compared to vertices of the cycle of  $\mathcal{C}$ . By construction of the orientation of  $M_0 \cup M$ , see Section 2.1, the orientation of the cycle is compatible with that of the edge  $(i_1, i_2)$ , where  $i_1$  is the smallest vertex of the cycle and  $i_2$  is its partner in  $M_0$ .

(IV)(3) If  $\ell_{k+1} \in \{\ell_1, \ell'_1, \dots, \ell_k, \ell'_k\}$ , then  $\gamma_{\ell_1}$  contains a loop of length  $\geq 3$ . If  $\ell_{k+1} = \ell_i$  for some  $i \in \{1, \dots, k\}$ , then the loop has even length and is alternating and the part of  $\gamma_{\ell_1}$  to the loop also has even length, is alternating and starts with an edge of  $M_0$ . Moreover, the orientation of the loop is compatible with the orientation of the edge  $(\ell_i, \ell'_i)$ , and the vertex  $\ell_1$  is smaller than all vertices of the path to the cycle and smaller than all vertices of the cycle.

Note that if  $\ell_{k+1} \neq \ell_1$ , then  $\ell_1$  is a leaf and the connected component containing  $\ell_1$  is a unicycle with a unique branch. Else, if  $\ell_{k+1} = \ell_1$ , the connected component is a cycle.

If  $\ell_{k+1} = \ell'_i$  for some  $i \in \{1, \dots, k\}$ , then the loop has odd length with two edges of  $M$  incident to the vertex  $\ell'_i$ . Observing that the loop in both directions is obtained from Step 1 of the algorithm, and using the fact that the matrix  $A$  is skew-symmetric, we deduce that the contributions of these configurations cancel and we remove them from the output of Step 1. Thus we only consider configurations such that  $\ell_{k+1} = \ell_i$  for some  $i \in \{1, \dots, k\}$ .

EXAMPLE. The output  $\mathcal{S}_1$  of Step 1 is:  $\mathcal{S}_1 = \{M_{1,5}, M_{1,2,1}, M_{1,2,4}, M_{1,2,5}, M_{1,3,1}, M_{1,3,4}\}$ . Configurations  $M_{1,5}$ ,  $M_{1,2,5}$  are in Case (IV)(1), configurations  $M_{1,2,1}$ ,  $M_{1,3,1}$  are in Case (IV)(3) with even cycles created, and configurations  $M_{1,2,4}$ ,  $M_{1,3,4}$  are in Case (IV)(3) with odd cycles created. Contributions of  $M_{1,2,4}$  and  $M_{1,3,4}$  cancel so that they are removed from the output. As a consequence the final output of Step 1 is, see also Figure 7:

$$\mathcal{S}_1 = \{M_{1,5}, M_{1,2,1}, M_{1,2,5}, M_{1,3,1}\},$$

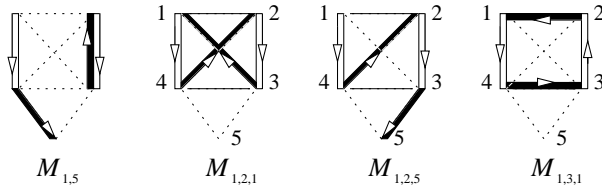


Figure 7: Output of Step 1 of the algorithm.

#### 2.4.2 Weight of configurations

As a consequence of the next two lemmas, we obtain that Step 1 of the algorithm is weight preserving *i.e.*  $w(\mathcal{S}_1) = w(M)$ , see Corollary 7.

**Lemma 5.** *We have:*

- $w(M) = \sum_{\ell_2 \in V_{\ell_1}} w(M_{\ell_1, \ell_2}).$

- If  $m \geq 2$ , then for every  $k \in \{2, \dots, m\}$  and every  $\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}, \dots, \ell_k \in V_{\ell_{k-1}}^{\mathcal{D}_{\ell_1}, \dots, \ell_{k-1}}$ :

$$w(M_{\ell_1, \dots, \ell_k}) = \sum_{\ell_{k+1} \in V_{\ell_k}} w(M_{\ell_1, \dots, \ell_{k+1}}).$$

*Proof.* Suppose that  $m \geq 2$ , the proof in the other case being similar. For every  $\ell_{k+1} \in V_{\ell_k}$ ,  $M_{\ell_1, \dots, \ell_{k+1}} = \{M_{\ell_1, \dots, \ell_k} \setminus (\ell'_k, \ell_k)\} \cup \{(\ell'_k, \ell_{k+1})\}$ , thus:

$$\prod_{e \in M_{\ell_1, \dots, \ell_{k+1}}} a_e = \frac{a_{\ell'_k, \ell_{k+1}}}{a_{\ell'_k, \ell_k}} \prod_{e \in M_{\ell_1, \dots, \ell_k}} a_e.$$

By assumption, coefficients of each line of the matrix  $A^R$  sum to 0. Returning to the definition of  $V_{\ell_k}$ , this implies that  $\sum_{\ell_{k+1} \in V_{\ell_k}} a_{\ell'_k, \ell_{k+1}} = -a_{\ell'_k, \ell_k}$ . Thus,

$$\sum_{\ell_{k+1} \in V_{\ell_k}} \prod_{e \in M_{\ell_1, \dots, \ell_{k+1}}} a_e = - \prod_{e \in M_{\ell_1, \dots, \ell_k}} a_e. \quad (3)$$

Combining Equation (3) with the definition of the weight of configurations given in Equation (2) yields:

$$\begin{aligned} \sum_{\ell_{k+1} \in V_{\ell_k}} w(M_{\ell_1, \dots, \ell_{k+1}}) &= \text{sgn}(\sigma_{M_0(M_{\ell_1, \dots, \ell_k}))} (-1)^{|C|} (-1)^{|\mathcal{D}|-k} \sum_{\ell_{k+1} \in V_{\ell_k}} \prod_{e \in M_{\ell_1, \dots, \ell_{k+1}}} a_e \\ &= \text{sgn}(\sigma_{M_0(M_{\ell_1, \dots, \ell_k}))} (-1)^{|C|} (-1)^{|\mathcal{D}|-(k-1)} \prod_{e \in M_{\ell_1, \dots, \ell_k}} a_e \\ &= w(M_{\ell_1, \dots, \ell_k}). \end{aligned}$$

□

**Lemma 6.** Suppose  $m \geq 2$ . Then for every  $k \in \{2, \dots, m\}$ ,

$$\sum_{i=k}^m w(\mathcal{O}_i) = \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}} \dots \sum_{\ell_k \in V_{\ell_{k-1}}^{\mathcal{D}_{\ell_1}, \dots, \ell_{k-1}}} w(M_{\ell_1, \dots, \ell_k}).$$

*Proof.* In order to simplify notations let us write, only in this proof,  $V_{\ell_k}^{\mathcal{D}}$  instead of  $V_{\ell_k}^{\mathcal{D}_{\ell_1}, \dots, \ell_k}$ . Lemma 6 is proved by backward induction on  $k$ .

Suppose  $k = m$ . By definition of the last step of the algorithm,  $V_{\ell_m}^{\mathcal{D}} = \emptyset$ , so that  $(V_{\ell_m}^{\mathcal{D}})^c = V_{\ell_m}$  and:

$$\begin{aligned} w(\mathcal{O}_m) &= \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \dots \sum_{\ell_m \in V_{\ell_{m-1}}^{\mathcal{D}}} \sum_{\ell_{m+1} \in V_{\ell_m}} w(M_{\ell_1, \dots, \ell_{m+1}}) \\ &= \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \dots \sum_{\ell_m \in V_{\ell_{m-1}}^{\mathcal{D}}} w(M_{\ell_1, \dots, \ell_m}), \text{ (by Lemma 5),} \end{aligned}$$

thus proving the case  $k = m$ . Suppose that the statement is true for some  $k \in \{3, \dots, m\}$ . By Iteration  $k - 1$  of Step 1 of the algorithm, we know that:

$$w(\mathcal{O}_{k-1}) = \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \cdots \sum_{\ell_{k-1} \in V_{\ell_{k-2}}^{\mathcal{D}}} \sum_{\ell_k \in (V_{\ell_{k-1}}^{\mathcal{D}})^c} w(M_{\ell_1, \dots, \ell_k})$$

Combining this with the induction hypothesis yields:

$$\begin{aligned} \sum_{i=k-1}^m w(\mathcal{O}_i) &= w(\mathcal{O}_{k-1}) + \sum_{i=k}^m w(\mathcal{O}_i) \\ &= \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \cdots \sum_{\ell_{k-1} \in V_{\ell_{k-2}}^{\mathcal{D}}} \left( \sum_{\ell_k \in (V_{\ell_{k-1}}^{\mathcal{D}})^c} + \sum_{\ell_k \in V_{\ell_{k-1}}^{\mathcal{D}}} \right) w(M_{\ell_1, \dots, \ell_k}) \\ &= \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \cdots \sum_{\ell_{k-1} \in V_{\ell_{k-2}}^{\mathcal{D}}} \sum_{\ell_k \in V_{\ell_{k-1}}^{\mathcal{D}}} w(M_{\ell_1, \dots, \ell_k}) \\ &= \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}}} \cdots \sum_{\ell_{k-1} \in V_{\ell_{k-2}}^{\mathcal{D}}} w(M_{\ell_1, \dots, \ell_{k-1}}) \text{ (by Lemma 5),} \end{aligned}$$

proving the statement for  $k - 1$  and ending the proof of Lemma 6.  $\square$

**Corollary 7.**

$$w(\mathcal{S}_1) = w(M).$$

*Proof.* Suppose  $m \geq 1$ . Then,

$$\begin{aligned} w(\mathcal{S}_1) &= w(\mathcal{O}_1) + \sum_{k=2}^m w(\mathcal{O}_k) \\ &= w(\mathcal{O}_1) + \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}} w(M_{\ell_1, \ell_2}), \text{ (by Lemma 6)} \\ &= \sum_{\ell_2 \in (V_{\ell_1}^{\mathcal{D}_{\ell_1}})^c} w(M_{\ell_1, \ell_2}) + \sum_{\ell_2 \in V_{\ell_1}^{\mathcal{D}_{\ell_1}}} w(M_{\ell_1, \ell_2}), \text{ (by definition of } \mathcal{O}_1) \\ &= \sum_{\ell_2 \in V_{\ell_1}} w(M_{\ell_1, \ell_2}) \\ &= w(M), \text{ (by Lemma 5).} \end{aligned}$$

When  $m = 0$ ,  $\mathcal{S}_1 = \{M\}$ , and the conclusion is immediate.  $\square$

## 2.5 Full algorithm

Let  $M_0$  be a reference perfect matching of the graph  $G$  and let  $M$  be a generic one. Recall that  $\mathcal{C}$  denotes the set of cycles of length  $\geq 4$  of the superimposition  $M_0 \cup M$ , and  $\mathcal{D}$  denotes its set of doubled edges.



In Section 2.4, we established Step 1 of the algorithm, starting from the superimposition  $M_0 \cup M$ , yielding a set of oriented edge configurations  $\mathcal{S}_1$  through the opening of doubled edges procedure, whose total weight is equal to the contribution of  $M$  to the Pfaffian. In this section, we introduce the full algorithm, which in essence consists of iterations of Step 1 performed until no doubled edges of  $\mathcal{D}$  remain.

Let us directly handle the following trivial case. If  $M_0 \cup M$  consists of cycles only, that is if the set  $\mathcal{D}$  is the empty, then the opening of edges procedure does not start, and recall that the output of Step 1 is  $\mathcal{S}_1 = \{M\}$ . The same holds for the full algorithm and its output is  $\mathcal{T} = \{M\}$ .

### 2.5.1 Step 1 of the full algorithm

Assume that the initial superimposition contains at least one doubled edge, *i.e.*  $\mathcal{D} \neq \emptyset$ .

Notations are complicated by the fact that the algorithm depends on the labeling of the vertices and that iterations of Step 1 depend on previous steps. We thus need many indices to keep track of everything rigorously, but one should keep in mind that, in essence, we are iterating Step 1. Let us add sub/superscripts to Step 1 of Section 2.4. That is  $\ell_1$  becomes  $\ell_1^1$ , Iteration  $k$  becomes  $k_1$  and Step 1 ends at time  $m_1$ . The set of configurations obtained from Step 1 is,  $\mathcal{S}_1 = \bigcup_{\gamma_{\ell_1^1} \in \Gamma_{\ell_1^1}} M_{\gamma_{\ell_1^1}}$ , and its weight is  $w(\mathcal{S}_1) = \sum_{\gamma_{\ell_1^1} \in \Gamma_{\ell_1^1}} w(M_{\gamma_{\ell_1^1}})$ .

For every  $\gamma_{\ell_1^1} \in \Gamma_{\ell_1^1}$ , let  $\mathcal{D}_{\gamma_{\ell_1^1}}$  be the set of doubled edges of the superimposition  $M_0 \cup M_{\gamma_{\ell_1^1}}$ . If  $\mathcal{D}_{\gamma_{\ell_1^1}} = \emptyset$ , then stop; else go to Step 2.

Output of Step 1 of the full algorithm. It is the subset  $\mathcal{T}_1$  of  $\mathcal{S}_1$ , consisting of configurations  $M_{\gamma_{\ell_1^1}}$  where  $\gamma_{\ell_1^1} \in \Gamma_{\ell_1^1}$ , and  $\mathcal{D}_{\gamma_{\ell_1^1}}$  is empty. Formally,

$$\mathcal{T}_1 = \bigcup_{\gamma_{\ell_1^1} \in (\tilde{\Gamma}_{\ell_1^1})^c} M_{\gamma_{\ell_1^1}},$$

where  $\tilde{\Gamma}_{\ell_1^1}$  is the set of paths  $\gamma_{\ell_1^1}$  of  $\Gamma_{\ell_1^1}$  such that  $\mathcal{D}_{\gamma_{\ell_1^1}}$  is non-empty. If for all  $\gamma_{\ell_1^1} \in \Gamma_{\ell_1^1}$ , the set  $\mathcal{D}_{\gamma_{\ell_1^1}}$  is non empty, then  $\mathcal{T}_1 = \emptyset$ .

EXAMPLE. Recall that the output of Step 1 is  $\mathcal{S}_1 = \{M_{1,5}, M_{1,2,1}, M_{1,2,5}, M_{1,3,1}\}$ . The set of doubled edges of the superimposition of  $M_0$  and  $M_{1,2,1}, M_{1,2,5}, M_{1,3,1}$  is empty, so that the output of Step 1 of the full algorithm is,  $\mathcal{T}_1 = \{M_{1,2,1}, M_{1,2,5}, M_{1,3,1}\}$ , and the algorithm continues with the configuration  $M_{1,5}$ .

### 2.5.2 Step $j$ of the full algorithm, $j \geq 2$

For every  $\gamma_{\ell_1^1} \in \tilde{\Gamma}_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}} \in \tilde{\Gamma}_{\ell_1^{j-1}}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-2}})$ , perform Step 1 of the algorithm with the initial superimposition  $M_0 \cup M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}}$ . That is, define  $\ell_1^j = \min\{i \in V : i \text{ belongs to a doubled edge of } \mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}}\}$ , and iterate until the algorithm ends at some

time  $m_j$ . Everything works out in the same way because  $\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}}$  is a subset of  $\mathcal{D}$ . The output is the set of oriented edge configurations:

$$\mathcal{S}_j(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}) = \bigcup_{\gamma_{\ell_1^j} \in \Gamma_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})} M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}},$$

where

$$\begin{aligned} \Gamma_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}) = & \left\{ \gamma_{\ell_1^j} : \gamma_{\ell_1^j} \text{ is a path } \ell_1^j, \ell_1^{j'}, \dots, \ell_{k_j}^j, \ell_{k_j}^{j'}, \ell_{k_j+1}^j, \text{ for some } k_j \in \{1, \dots, m_j\}, \right. \\ & \text{such that } \ell_1^j = \min\{i \in V : i \text{ belongs to a doubled edge of } \mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}}\} \\ & \forall i \in \{2, \dots, k_j\}, \ell_i^j \in V_{\ell_{i-1}^j}^{\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}, \ell_1^j, \dots, \ell_{i-1}^j}}, \\ & \text{and } \ell_i^{j'} \text{ is the partner of } \ell_i^j \text{ in } M_0 \text{ and } M, \\ & \left. \ell_{k_j+1}^j \in \left( V_{\ell_{k_j}^j}^{\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}, \ell_1^j, \dots, \ell_{k_j}^j}} \right)^c \right\}. \end{aligned} \quad (4)$$

$$M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}} = M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}, \ell_1^j, \dots, \ell_{k_j+1}^j},$$

$$w(M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}) = \text{sgn}(\sigma_{M_0}(M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}, \ell_1^j, \dots, \ell_{k_j}^j})) (-1)^{|\mathcal{D}| - (k_1 + \dots + k_j)} (-1)^{|\mathcal{C}|} \prod_{e \in M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}} a_e.$$

For every  $\gamma_{\ell_1^j} \in \Gamma_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})$ , do the following: if  $\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}$  is empty, stop; else go to step  $j+1$ .

Output of Step  $j$  of the full algorithm.

Let  $\mathcal{T}_j(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})$  be the subset of  $\mathcal{S}_j(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})$ , consisting of configurations  $M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}$  such that  $\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}$  is empty. Formally,

$$\mathcal{T}_j(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}) = \bigcup_{\gamma_{\ell_1^j} \in (\tilde{\Gamma}_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}))^c} M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}},$$

where  $\tilde{\Gamma}_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})$  is the subset of paths  $\gamma_{\ell_1^j}$  of  $\Gamma_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}})$  such that  $\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}$  is non-empty. If  $\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}$  is non empty, then  $\mathcal{T}_j(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}) = \emptyset$ .

Then, the output  $\mathcal{T}_j$  of Step  $j$  of the full algorithm is:

$$\mathcal{T}_j = \bigcup_{\gamma_{\ell_1^1} \in \tilde{\Gamma}_{\ell_1^1}} \dots \bigcup_{\gamma_{\ell_1^{j-1}} \in \tilde{\Gamma}_{\ell_1^{j-1}}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-2}})} \bigcup_{\gamma_{\ell_1^j} \in (\tilde{\Gamma}_{\ell_1^j}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{j-1}}))^c} M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}.$$

For convenience, we shall also use the notation  $\Gamma_j$  for the paths  $(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j})$  involved in  $\mathcal{T}_j$ , i.e:

$$\mathcal{T}_j = \bigcup_{(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}) \in \Gamma_j} M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}.$$

The weight of  $\mathcal{T}_j$  is the sum of the weights of the configurations it contains.

EXAMPLE. In Step 2, we perform Step 1 of the algorithm starting from the initial superimposition  $M_0 \cup M_{1,5}$ . The latter contains one doubled edge 23, thus the vertex  $\ell_1^2$  is the smallest of 2 and 3, that is 2. The output  $\mathcal{S}_2$  consists of the configurations  $M_{1,5;2,1}, M_{1,5;2,4}, M_{1,5;2,5}$ , depicted in Figure 8 below.

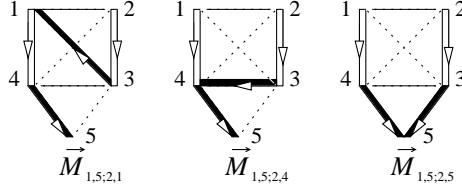


Figure 8: Output of Step 2 of the algorithm.

The superimposition of  $M_0$  and the above three configurations contains no doubled edges. As a consequence, the full algorithm stops and the output  $\mathcal{T}_2$  of Step 2 is:

$$\mathcal{T}_2 = \{M_{1,5;2,1}, M_{1,5;2,4}, M_{1,5;2,5}\}.$$

### 2.5.3 End and output of the full algorithm

The algorithm stops at Step  $T$  for the first time, if it hasn't stopped at time  $T - 1$ , and if for every  $\gamma_{\ell_1^1} \in \tilde{\Gamma}_{\ell_1^1}, \dots, \gamma_{\ell_1^{T-1}} \in \tilde{\Gamma}_{\ell_1^{T-1}}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{T-2}})$ ,  $\gamma_{\ell_1^T} \in \Gamma_{\ell_1^T}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{T-1}})$ , the superimposition  $M_0 \cup M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^T}}$  contains no doubled edge. This implies in particular that  $(\tilde{\Gamma}_{\ell_1^T}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{T-1}}))^c = \Gamma_{\ell_1^T}(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{T-1}})$ . Since the number of doubled edges decreases at every step and since  $\mathcal{D}$  is finite, we are sure that this happens in finite time.

The output  $\mathcal{T}$  of the full algorithm is :

$$\mathcal{T} = \bigcup_{j=1}^T \mathcal{T}_j.$$

EXAMPLE. The output of the full algorithm is:

$$\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 = \{M_{1,2,1}, M_{1,2,5}, M_{1,3,1}, M_{1,5;2,1}, M_{1,5;2,4}, M_{1,5;2,5}\},$$

summarized in Figure 9 below.

The *weight* of  $\mathcal{T}$  is the sum of the weights of the configurations it contains. If the initial superimposition  $M_0 \cup M$  consists of cycles only, *i.e.* the set  $\mathcal{D}$  is empty, then  $\mathcal{T} = \{M\}$ , and

$$w(\mathcal{T}) = w(M) = \text{sgn}(\sigma_{M_0(M)})(-1)^{|\mathcal{C}|} \prod_{e \in M} a_e.$$

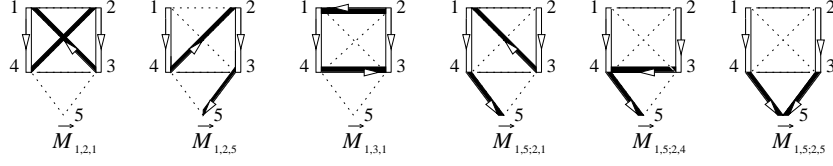


Figure 9: Output of the full algorithm.

In all other cases:

$$w(\mathcal{T}) = \sum_{j=1}^T \sum_{(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}) \in \Gamma_j} w(M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}).$$

Since for every  $j \in \{1, \dots, T\}$ , and for every  $(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}) \in \Gamma_j$ , the superimposition  $M_0 \cup M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}$  contains no doubled edge of  $\mathcal{D}$ , we have:

$$w(M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}) = \text{sgn}(\sigma_{M_0(M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}})}) (-1)^{|\mathcal{C}|} \prod_{e \in M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}} a_e. \quad (5)$$

By iterating the argument of Section 2.4.2, we obtain the following:

**Corollary 8.** *The full algorithm is weight preserving, that is:*

$$w(\mathcal{T}) = w(M).$$

## 2.6 Geometric characterization of configurations

Consider the superimposition  $M_0 \cup M$ , recall that  $\mathcal{C}$  denotes the set of cycles of length  $\geq 4$  of  $M_0 \cup M$ , and that  $\mathcal{D}$  denotes its set of doubled edges. Consider the full algorithm with initial superimposition  $M_0 \cup M$  in the case where  $M_0 \cup M$  contains doubled edges, that is when  $\mathcal{D} \neq \emptyset$ . Let  $j \in \{1, \dots, T\}$ ,  $(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}) \in \Gamma_j$ , and  $M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}} \in \mathcal{T}$  be a generic output; and denote by  $F_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}$  the superimposition  $M_0 \cup M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}}$ . In order to simplify notations, we introduce:

$$\forall i \in \{1, \dots, j\}, \quad F_i := F_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_i^i}}.$$

One should keep in mind that the index  $j$  refers to the last step of the algorithm, and that indices  $i \in \{1, \dots, j-1\}$  refer to intermediate steps. As a consequence of the algorithm, see (4), the configuration  $F_j$  and the paths  $\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}$  satisfy the following properties.

- The oriented edge configuration  $F_j$ :
  - (I) has one outgoing edge at every vertex of  $V$ . It consists of the paths  $\gamma_{\ell_1^1}, \dots, \gamma_{\ell_j^j}$  and of the cycles  $\mathcal{C}$  of the initial superimposition  $M_0 \cup M$ ;
  - (II) has no doubled edge of  $\mathcal{D}$  since the full algorithm precisely stops when this is the case.

For every  $i \in \{1, \dots, j\}$ , the path  $\gamma_{\ell_1^i}$  satisfies the following.

- (III) It has even length  $2k_i$  for some  $k_i \in \{1, \dots, m_i\}$  and is alternating. It starts from the vertex  $\ell_1^i$ , followed by an edge of  $M_0$ .
- (IV) The vertex  $\ell_1^i$  is the smallest vertex belonging to a doubled edge of  $F_{i-1}$  (understood as  $M_0 \cup M$  when  $i = 1$ ). The  $2k_i$  first vertices are all distinct and the last vertex  $\ell_{k_i+1}^i$  belongs to:

$$\left( V_{\ell_{k_i}^i}^{\mathcal{D}_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{i-1}}, \ell_1^i, \dots, \ell_{k_i}^i}} \right)^c = V_{\ell_{k_i}^i} \cap \{R \cup V^{\mathcal{C}} \cup V^{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{i-1}}} \cup \{\ell_1^i, (\ell_1^i)', \dots, \ell_{k_i}^i, (\ell_{k_i}^i)'\}\}.$$

As a consequence, one of the following 5 cases hold.

- If  $\ell_{k_i+1}^i \in R \cup V^{\mathcal{C}} \cup \{\ell_1^i, (\ell_1^i)', \dots, \ell_{k_i}^i, (\ell_{k_i}^i)'\}$ , then  $\gamma_{\ell_1^i}$  consists of a new connected component, and we recover the three cases obtained after Step 1 of the algorithm, replacing  $\gamma_{\ell_1}$  by  $\gamma_{\ell_1^i}$ , see Section 2.4.1. For convenience of the reader, we repeat these cases here.

- (IV)(1) If  $\ell_{k_i+1}^i \in R$ , then  $\gamma_{\ell_1^i}$  is a loopless oriented path from  $\ell_1^i$  to one of the root vertices of  $R$ . Since  $R = \{n+1, \dots, n+r\}$ ,  $\ell_1^i$  is smaller than all vertices of  $\gamma_{\ell_1^i}$ . The vertex  $\ell_1^i$  is a leaf of a connected component of  $F_i$ , which consists of the path  $\gamma_{\ell_1^i}$ .
- (IV)(2) If  $\ell_{k_i+1}^i \in V^{\mathcal{C}}$ , then  $\gamma_{\ell_1^i}$  is a loopless oriented path ending at a vertex of one of the cycles of  $\mathcal{C}$ . The vertex  $\ell_1^i$  is smaller than the  $2k_i$  first vertices of the path, but cannot be compared to vertices of the cycle of  $\mathcal{C}$ . By construction of the orientation of  $M_0 \cup M$ , see Section 2.1, the orientation of the cycle is compatible with that of the edge  $(i_1, i_2)$ , where  $i_1$  is the smallest vertex of the cycle and  $i_2$  is its partner in  $M_0$ . The vertex  $\ell_1^i$  is a leaf of a connected component of  $F_i$ , which is a unicycle with  $\gamma_{\ell_1^i}$  as unique branch and a cycle of  $\mathcal{C}$  as cycle.
- (IV)(3) when  $\ell_{k_i+1}^i \in \{\ell_1^i, (\ell_1^i)', \dots, \ell_{k_i}^i, (\ell_{k_i}^i)'\}$ , then  $\gamma_{\ell_1^i}$  contains a loop of length  $\geq 3$ . Recall that configurations with odd cycles cancel because of the skew-symmetry of the matrix, so that we only consider configurations where  $\ell_{k_i+1}^i = \ell_s^i$  for some  $s \in \{1, \dots, k_i\}$ . In this case, the part of the path  $\gamma_{\ell_1^i}$  to the loop has even length, is alternating and start with an edge of  $M_0$ . The loop has even length  $\geq 4$ , is alternating and its orientation is compatible with the orientation of the edge  $(\ell_s^i, \ell_s^{i'})$ . The vertex  $\ell_1^i$  is smaller than all vertices of the path to the cycle and smaller than all vertices of the cycle.  
 If  $\ell_{k_i+1}^i \neq \ell_1^i$ , then  $\ell_1^i$  is a leaf of a connected component of  $F_i$  which is a unicycle with a unique branch, consisting of the path  $\gamma_{\ell_1^i}$ .  
 If  $\ell_{k_i+1}^i = \ell_1^i$ , then  $\ell_1^i$  is the smallest vertex of a connected component of  $F_i$  which is a cycle, consisting of the path  $\gamma_{\ell_1^i}$ .

- If  $\ell_{k_i+1}^i \in V^{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^{i-1}}}$  then, the path  $\gamma_{\ell_1^i}$  attaches itself to a connected component of  $F_{i-1}$ , this can only occur when  $i \geq 2$ , and one of the following happens.

- (IV)(4) The path  $\gamma_{\ell_1^i}$  attaches itself to a leaf of  $F_{i-1}$ , that is  $\ell_{k_j+1} = \ell_1^t$  for some  $t \in \{1, \dots, i-1\}$ . Then  $\gamma_{\ell_1^i}$  is a loopless oriented path from  $\ell_1^i$  to  $\ell_1^t$ . The vertex  $\ell_1^i$  is smaller than the  $2k_i$  following ones, but greater than  $\ell_1^t$ . Indeed  $\ell_1^t$  is the starting point of a previous step of the algorithm. *This allows to identify the ending vertex of the path  $\gamma_{\ell_1^i}$ .* The vertex  $\ell_1^i$  is a leaf of  $F_i$ .
- (IV)(5) The path  $\gamma_{\ell_1^i}$  creates a new branch of the component. Then  $\gamma_{\ell_1^i}$  is a loopless oriented path. The vertex  $\ell_1^i$  is smaller than the  $2k_i$  following ones, but we have no a priori information on the last vertex of the path. The last vertex of the path  $\gamma_{\ell_1^i}$  is nevertheless identified as being a fork. The vertex  $\ell_1^i$  is a leaf of  $F_i$ . Note that the component of  $\ell_1^i$  might be a unicycle with a unique branch. If this is the case, the branch is the path  $\gamma_{\ell_1^i}$  and the cycle was created by Case (IV)(3) in a previous step of the algorithm. The vertex  $\ell_1^i$  is thus larger than the smallest vertex of the cycle.

**Lemma 9.** *The oriented edge configuration  $F_j$  is an RCRSF compatible with  $M_0$ .*

*Proof.* By definition of the algorithm, the oriented edge configuration  $F_j$  contains as many edges as  $M_0 \cup M$ , that is  $|V|$  edges. By definition, it contains all edges of  $M_0$ , that is  $|V|/2$  edges, and by the algorithm no doubled edges of  $\mathcal{D}$ , that is  $|V|/2$  edges not in  $M_0$ .

By Point (I) the oriented edge configuration  $F_j$  has one outgoing edge at every vertex of  $V$ , which is equivalent to saying that it is an RCRSF such that edges of each component are oriented towards its root, and cycles are oriented in one of the two possible directions. It thus remains to show that cycles of unicycles are alternating, and have even length  $\geq 4$ . By Point (II), the oriented edge configuration  $F_j$  has no doubled edge of  $\mathcal{D}$ , thus if  $F_j$  has a cycle, it either comes from Point (IV)(2) meaning that it is a cycle of  $\mathcal{C}$  implying that it is even, alternating and has length  $\geq 4$ ; or from Point (IV)(3), when it is created by the algorithm. Returning to the description of Point (IV)(3) and recalling that the contribution of configurations with odd cycles cancel, we know that it has the same properties in this case.  $\square$

For every  $i \in \{1, \dots, j\}$ , and for every connected component of  $F_i$  which is a cycle  $C$  created by the algorithm (*i.e.* not a cycle of the initial superimposition), denote by  $m_C$  the smallest vertex of  $C$ . Define  $x_i$  to be:

$$x_i = \begin{cases} \max\{m_C : C \text{ is a cycle-connected component of } F_i, \text{ but not of } \mathcal{C}\} & \text{if } \{ \} \neq \emptyset \\ -\infty & \text{else.} \end{cases}$$

If  $F_i$  has at least one leaf, let  $y_i$  be the maximum leaf of  $F_i$ , else let  $y_i = -\infty$ .

If both  $x_i$  and  $y_i$  are  $-\infty$ , then  $F_i$  has no leaves and only contains cycles of the initial superimposition  $M_0 \cup M$ . This means that the set  $\mathcal{D}$  is empty, and that  $F$  is the initial superimposition  $M_0 \cup M$ . This has been excluded here, since the full algorithm doesn't even start the opening of edges procedure in this case. Thus  $\max\{x_i, y_i\} > -\infty$ .

**Lemma 10.** *For every  $i \in \{1, \dots, j\}$ , the initial vertex  $\ell_1^i$  of Step  $i$  is the maximum of  $x_i$  and  $y_i$ .*

*Proof.* By Point (IV) above, the vertex  $\ell_1^i$  is either a leaf of  $F_i$  or the smallest vertex of a connected component of  $F_i$  which is a cycle created by the algorithm, meaning that it is not a cycle of  $\mathcal{C}$  i.e. not a cycle of the initial superimposition  $M_0 \cup M$ . Arguing by induction, all leaves and smallest vertices of cycle-components of  $F_i$  which are not present in  $\mathcal{C}$ , must be initial vertices of steps  $i$  of the algorithm for some  $i \in \{1, \dots, j\}$ . Moreover by construction, the vertex  $\ell_1^i$  is larger than all previous initial steps of the algorithm, thus proving the lemma.  $\square$

Properties described in Point(IV) also characterize the path  $\gamma_{\ell_1^i}$  once the initial vertex  $\ell_1^i$  is fixed. This can be summarized in the following lemma.

**Lemma 11.** *Let  $\ell_1^i$  be the initial vertex of Step  $i$ . Then,*

- *Suppose that  $\ell_1^i$  is a leaf of a connected component of  $F_i$ . When the connected component is a unicycle rooted on a cycle created by the algorithm, we assume moreover that it contains more than one branch. Then, we are in Cases (IV)(1)(2) or (5) and the path  $\gamma_{\ell_1^i}$  is characterized as the subpath of the unique path from  $\ell_1^i$  to the root of the connected component, stopping the first time one visits a vertex which: belongs to  $R$  or to the cycle of the component; is a fork; is smaller than  $\ell_1^i$ .*
- *Suppose that  $\ell_1^i$  is the leaf of a unicycle of  $F_i$  rooted on a cycle created by the algorithm and containing a unique branch. If  $\ell_1^i$  is larger than the smallest vertex of the cycle, then we are in Case (IV)(5) and the path  $\gamma_{\ell_1^i}$  is the path from  $\ell_1^i$  to the cycle, stopping when the cycle is reached. Else, if  $\ell_1^i$  is smaller than the smallest vertex of the cycle, we are in Case (IV)(3) and the path  $\gamma_{\ell_1^i}$  is the path from  $\ell_1^i$  to the cycle, followed by the cycle, with the orientation specified in (IV)(3).*
- *If  $\ell_1^i$  is the smallest vertex of a connected component of  $F_i$  which is a cycle created by the algorithm, then we are in Case (IV)(3) and the path  $\gamma_{\ell_1^i}$  is the cycle, with the orientation specified in (IV)(3).*

*Remark 12.* If the initial superimposition  $M_0 \cup M$  consists of cycles only, that is if the set  $\mathcal{D}$  is empty, then the output of the full algorithm is  $F = M_0 \cup M$ , which consists of alternating cycles of even length  $\geq 4$ . The orientation of cycles is specified in Section 2.1, and cycles are oriented in one of the possible two directions. In this case also,  $F$  is an RCRSF compatible with  $M_0$ .

### 3 Proofs and corollaries

We now prove Theorem 1, Corollary 3 and state and prove the line bundle version of the result.

#### 3.1 Proof of Theorem 1

In this section, we prove Theorem 1. Let  $M_0$  be a reference perfect matching of  $G$ , and let  $F$  be an RCRSF compatible with  $M_0$ , containing  $k_F$  unicycles. In Lemma 14, we suppose

that  $F$  is an output of the full algorithm and identify  $2^{k_F}$  possible perfect matchings  $M$  for the initial superimposition  $M_0 \cup M$ . Then, we introduce a partial reverse algorithm used to define Condition (C) for RCRSFs compatible with  $M_0$ . In Proposition 15, we prove that an RCRSF compatible with  $M_0$  is an output of the full algorithm if and only if it satisfies Condition (C), and if this is the case, it is obtained  $2^{k_F}$  times. The remainder of the proof consists in showing that contribution of RCRSFs containing unicycles cancel, and that only spanning forests remain with the appropriate weight, thus proving Theorem 1.

Let  $F$  be an RCRSF compatible with  $M_0$ , and let  $k_F$  denote the number of unicycles it contains. If  $k_F \neq 0$ , we let  $\{C_1, \dots, C_{k_F}\}$  be its set of cycles. For every  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ , define the edge configuration  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  as follows:

$$M^{(\varepsilon_1, \dots, \varepsilon_{k_F})} = \begin{cases} \text{edges of } M_0 \text{ on branches of } F \\ \text{edges of } M_0 \text{ on the cycle } C_j, \text{ when } \varepsilon_j = 0 \\ \text{edge of } F \setminus M_0 \text{ on the cycle } C_j, \text{ when } \varepsilon_j = 1. \end{cases}$$

If  $k_F = 0$ , then the set of cycles of  $F$  is  $\{\emptyset\}$ , and we set  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})} = M_0$ .

**Lemma 13.** *For every RCRSF  $F$  compatible with  $M_0$ , and every  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ , the edge configuration  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  is a perfect matching of  $G$ .*

*Proof.* If  $F$  contains no unicycles,  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})} = M_0$  and this is immediate. Suppose  $k_F \neq 0$ . Since  $M_0$  is a perfect matching, and since the restriction of  $M_0$  and the restriction of  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  to branches of  $F$  are the same, all vertices of  $V \setminus \{V(C_1), \dots, V(C_{k_F})\}$  are incident to exactly one edge of  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ . Moreover, by assumption for every  $j$ , the cycle  $C_j$  is alternating, implying that each vertex of  $V(C_j)$  is incident to exactly one edge of the restriction of  $M_0$  and one edge of the restriction of  $F \setminus M_0$  to  $C_j$ . As a consequence, every vertex of  $V$  is incident to exactly one edge of  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ , proving that it is a perfect matching of  $G$ .  $\square$

**Lemma 14.** *Let  $F$  be the superimposition of  $M_0$  and of an output of the full algorithm. Then, the perfect matching  $M$  of the initial superimposition  $M_0 \cup M$ , must be equal to  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  for some  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ .*

*Proof.* If  $F$  is an output of the full algorithm, then by Lemma 9 and Remark 12, it is an RCRSF compatible with  $M_0$ , so that  $\forall (\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ , the perfect matching  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  is well defined. Suppose that  $F$  is an output of the full algorithm with initial superimposition  $M_0 \cup M$ , where  $M$  is not  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  for some  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ . Then,  $M_0 \cup M$  contains at least one cycle  $C$  which is not  $C_1, \dots, C_{k_F}$ . Returning to the definition of the algorithm, we know that cycles present in the initial superimposition are also present in the output  $F$ . This yields a contradiction since  $F$  contains exactly the cycles  $C_1, \dots, C_{k_F}$ .  $\square$

### Partial reverse algorithm

Input: an RCRSF  $F$  compatible with  $M_0$  not consisting of cycles only.

Initialization:  $F_1 = F$ .



Step  $i, i \geq 1$

Let  $\bar{\ell}_1^i$  be the largest leaf of  $F_i$ , and consider the connected component containing  $\bar{\ell}_1^i$ . Start from  $\bar{\ell}_1^i$  along the unique path joining  $\bar{\ell}_1^i$  to the root or the cycle of the component, until the first time one of the following vertices is reached:

- the root vertex if the component is a tree, or the cycle if it is a unicycle;
- a fork;
- a vertex which is smaller than the leaf  $\bar{\ell}_1^i$ .

This yields a loopless path  $\lambda_{\bar{\ell}_1^i}$  starting from  $\bar{\ell}_1^i$ , of length  $\geq 1$ . Let  $F_{i+1} = F_i \setminus \lambda_{\bar{\ell}_1^i}$ . If  $F_{i+1}$  is empty or contains cycles only, then stop. Else, go to step  $i + 1$ .

End: since edges are removed at every step and since  $F$  contains finitely many edges, the algorithm ends in finite time  $N$ .

**Definition 3.1.** An RCRSF  $F$  compatible with  $M_0$  is said to satisfy *Condition (C)* if either  $F$  consists of cycles only, or if each of the paths  $\lambda_{\bar{\ell}_1^1}, \dots, \lambda_{\bar{\ell}_1^N}$ , obtained from the partial reverse algorithm has even length and starts from an edge of  $M_0$ .

**Proposition 15.** Let  $F$  be an RCRSF compatible with  $M_0$ .

Then, for every  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ ,  $F$  is the superimposition of  $M_0$  and of an output of the full algorithm, with initial superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ , if and only if  $F$  satisfies condition (C). The orientation of cycles of  $F$  is specified by the proof.

*Proof.* Let  $F$  be an RCRSF compatible with  $M_0$  containing  $k_F$  unicycles, and denote by  $\{C_1, \dots, C_{k_F}\}$  its set of cycles. For every  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ , the edge configuration  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  is well defined, and by Lemma 13 is a perfect matching. We now fix  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ . Recall that if  $k_F = 0$ , then the perfect matching  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  is simply  $M_0$ .

In the case where  $k_F \neq 0$ ,  $(\varepsilon_1, \dots, \varepsilon_{k_F}) = (1, \dots, 1)$ , and  $F$  consists of cycles only, the superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  consists of cycles only, and  $F = M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  is an output of the full algorithm. The orientation of the cycles is specified by the choice of orientation of Section 2.1, thus proving Proposition 15.

Assume that we are not in the above case. Then  $F$  is an output of the algorithm with initial superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  if and only if there exists a positive integer  $j$  and a sequence of paths  $(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}) \in \Gamma_j$  such that  $F = M_0 \cup M_{\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}}^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ .

Lemmas 10 and 11 give a characterization of  $\ell_1^i$  and  $\gamma_{\ell_1^i}$  at every step of the algorithm. This allows us to define a full reverse algorithm.

**Full reverse algorithm**

Input: an RCRSF  $F$  compatible with  $M_0$ ,  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$  as above, and the corresponding perfect matching  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ . If  $k_F = 0$ , then  $M^{(\varepsilon_1, \dots, \varepsilon_{k_F})} = M_0$ .

Initialization:  $F_1 = F$ .

Step  $i, i \geq 1$ .

Since  $F_i$  is either  $F_1$  or is obtained from  $F_{i-1}$  by removing edges, the set of cycles of  $F_i$  is included in the set of cycles  $\{C_1, \dots, C_{k_F}\}$  of  $F_1$ .

For every connected component of  $F_i$  which is a cycle  $C_\alpha$  such that  $\varepsilon_\alpha = 0$  (meaning that  $C_\alpha$  is not a cycle of the initial superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ ), let  $m_{C_\alpha}$  be the smallest vertex of  $C_\alpha$ . Define

$$x_i = \begin{cases} \max\{m_{C_\alpha} : C_\alpha \text{ is a cycle-connected component of } F_i, \text{ and } \varepsilon_\alpha = 0\} & \text{if } \{\} \neq \emptyset \\ -\infty & \text{else.} \end{cases}$$

If  $F_i$  has at least one leaf, let  $y_i$  be the maximum leaf, else let  $y_i = -\infty$ . Note that by assumption, we do not have  $x_i = y_i = -\infty$ . We let  $\ell_1^i = \max\{x_i, y_i\}$ , and  $\gamma_{\ell_1^i}$  be the oriented path as characterized in Lemma 11.

Let  $F_{i+1} = F_i \setminus \gamma_{\ell_1^i}$ . If the oriented edge configuration  $F_{i+1}$  is empty, or if it consists of cycles of the superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  only, then stop; else go to step  $i + 1$ .

End: since edges are removed at every step and since  $F$  contains finitely many edges, the algorithm ends in finite time  $j$ , for some integer  $j$ .

This defines for every RCRSF  $F$  compatible with  $M_0$ , a sequence of paths  $\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j}$  such that  $F$  is the union of these paths and of cycles of the initial superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ . As a consequence, the oriented edge configuration  $F$  satisfies Properties (I), (II), (IV)(1)–(5). We are thus left with proving that  $F$  satisfies Property (III) if and only if it satisfies Condition (C), *i.e.* we need to show that the paths  $(\gamma_{\ell_1^1}, \dots, \gamma_{\ell_1^j})$  all have even length, are alternating and start from an edge of  $M_0$  if and only if  $F$  satisfies Condition (C).

Observe that initial vertices  $(\ell_1^1, \dots, \ell_1^j)$  of the full reverse algorithm consist of initial vertices  $(\bar{\ell}_1^1, \dots, \bar{\ell}_1^N)$  of the partial reverse algorithm, interlaced with smallest vertices of components which are cycles. Indeed, the only difference in the partial reverse algorithm is that cycles are not removed, but this does not change the characterization of largest leaf.

If  $\ell_1^i$  is the smallest vertex of a component of  $F_i$  which is a cycle, that is if  $\ell_1^i = x_i$ , then  $\gamma_{\ell_1^i}$  is a cycle  $C_\alpha \in \{C_1, \dots, C_{k_F}\}$  such that  $\varepsilon_\alpha = 0$ . Since  $F$  is compatible with  $M_0$ , the cycle has even length and is alternating. The orientation is fixed by the algorithm and  $\gamma_{\ell_1^i}$  always satisfies Property (III).

If  $\ell_1^i$  is the largest leaf of  $F_i$ , that is if  $\ell_1^i = y_i$ , then in all cases except one, which we treat below, the path  $\gamma_{\ell_1^i}$  is exactly the path  $\lambda_{\bar{\ell}_1^{i'}}$  of the partial reverse algorithm, for some  $i' \leq i$ . Condition (C) says that  $\lambda_{\bar{\ell}_1^{i'}}$  has even length and starts from an edge of  $M_0$ . In order to show that this is equivalent to satisfying Property (III), we are left with showing that, by construction, the path  $\lambda_{\bar{\ell}_1^{i'}}$  is always alternating. Suppose that this is not the case, then there are at least two edges of the same kind (either in  $M_0$  or not in  $M_0$ ) which follow each other. This implies that there is a vertex  $v$  of the path incident to two edges of the same kind. Since  $M_0$  is a perfect matching, every vertex is incident to exactly one edge of  $M_0$ , so that we cannot have two edges of  $M_0$  following each other. Thus these two edges

do not belong to  $M_0$ . Again, since  $M_0$  is a perfect matching, the vertex  $v$  is also incident to an edge of  $M_0$ , implying that  $v$  is the end of a branch. By construction of the path  $\lambda_{\bar{\ell}_1'}$ , the path must stop at  $v$ , implying that one of the two edges is not in  $\lambda_{\bar{\ell}_1'}$ , yielding a contradiction.

We now treat the last case. If  $\ell_1^i$  is a leaf of a connected component of  $F_i$  which is a unicycle rooted on a cycle  $C_\alpha$  such that  $\varepsilon_\alpha = 0$ , with a unique branch, and such that  $\ell_1^i$  is smaller than the smallest vertex of the cycle. Then the path  $\gamma_{\ell_1^i}$  is the path  $\lambda_{\bar{\ell}_1'}$  followed by the cycle with the appropriate orientation. We have to show that  $\gamma_{\ell_1^i}$  satisfies Property (III) if and only if  $\lambda_{\bar{\ell}_1'}$  satisfies Condition (C). By Property (III), we know that the part of  $\gamma_{\ell_1^i}$  stopping when the cycle is reached, which is precisely  $\lambda_{\bar{\ell}_1'}$ , has even length and starts from an edge of  $M_0$ . This is exactly Condition (C), since by the same argument as above, the path  $\lambda_{\bar{\ell}_1'}$  is alternating. We conclude by observing that since  $F$  is compatible with  $M_0$ , the cycle part of  $\gamma_{\ell_1^i}$  is alternating, and starts from an edge of  $M_0$  by construction of the orientation of the cycle. Thus  $\gamma_{\ell_1^i}$  satisfies Property (III) if and only if  $F$  is compatible with  $M_0$  and satisfies Condition (C).  $\square$

We denote by  $\mathcal{G}(M_0)$  the set of RCRSFs compatible with  $M_0$  satisfying Condition (C). Let  $F$  be an RCRSF of  $\mathcal{G}(M_0)$ , and let  $k_F$  be its number of unicycles. If  $k_F \neq 0$ , then for every  $(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0, 1\}^{k_F}$ , denote by  $F^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  the version of  $F$  obtained from the full algorithm with initial superimposition  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$ , with the orientation of cycles given by Proposition 15. If  $k_F = 0$ , then  $F$  is obtained exactly once from the full algorithm with initial superimposition  $M_0 \cup M_0$ .

Since  $M_0 \cup M^{(\varepsilon_1, \dots, \varepsilon_{k_F})}$  has exactly  $\sum_{i=1}^{k_F} \varepsilon_i$  cycles, and since  $M_0 \cup M_0$  has none, we have as a consequence of the full algorithm, see Equation (5), that the weight  $w_{M_0}(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)$  is equal to:

$$\text{sgn}(\sigma_{M_0(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}) \cdot \begin{cases} (-1)^{\sum_{i=1}^{k_F} \varepsilon_i} \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e & \text{if } k_F \neq 0 \\ \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e & \text{if } k_F = 0. \end{cases} \quad (6)$$

Recall that  $\mathcal{T}$  denotes the output of the full algorithm with initial superimposition  $M_0 \cup M$  for a fixed reference perfect matching  $M_0$  and a generic perfect matching  $M$  of  $G$ . Since we now aim at taking the union over all perfect matchings  $M$ , we write  $\mathcal{T}$  as  $\mathcal{T}_{M_0}(M)$ .

As a consequence of Proposition 15, we have that  $\bigcup_{M \in \mathcal{M}} \mathcal{T}_{M_0}(M)$  is equal to:

$$\left( \bigcup_{\{F \in \mathcal{G}(M_0) : k_F \neq 0\}} \bigcup_{(\varepsilon_1, \dots, \varepsilon_k) \in \{0, 1\}^{k_F}} F^{(\varepsilon_1, \dots, \varepsilon_k)} \setminus M_0 \right) \bigcup \left( \bigcup_{\{F \in \mathcal{G}(M_0) : k_F = 0\}} F \setminus M_0 \right). \quad (7)$$

Returning to the definition of the Pfaffian of Equation (1), using Corollary 8 and Equation

(7) in the last line, we deduce that:

$$\begin{aligned}
\text{Pf}(A) &= \sum_{M \in \mathcal{M}} w_{M_0}(M), \text{ (Definition of Equation (1))} \\
&= \sum_{M \in \mathcal{M}} w(\mathcal{T}_{M_0}(M)), \text{ (by Corollary 8)} \\
&= \underbrace{\sum_{\{F \in \mathcal{G}(M_0) : k_F \neq 0\}} \sum_{(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0,1\}^{k_F}} w_{M_0}(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}_{(I)} + \underbrace{\sum_{\{F \in \mathcal{G}(M_0) : k_F = 0\}} w_{M_0}(F \setminus M_0)}_{(II)}.
\end{aligned}$$

Let us show that (I) is equal to zero. As a consequence of Equation (6), it is equal to:

$$(I) = \sum_{\{F \in \mathcal{G}(M_0) : k_F \neq 0\}} \sum_{(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0,1\}^{k_F}} \text{sgn}(\sigma_{M_0(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}) (-1)^{\sum_{i=1}^{k_F} \varepsilon_i} \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e.$$

Observing that the term  $\text{sgn}(\sigma_{M_0(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}) \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e$  is independent of  $(\varepsilon_1, \dots, \varepsilon_{k_F})$ , we conclude that:

$$\begin{aligned}
(I) &= \sum_{\{F \in \mathcal{G}(M_0) : k_F \neq 0\}} \left( \text{sgn}(\sigma_{M_0(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}) \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e \right) \sum_{(\varepsilon_1, \dots, \varepsilon_{k_F}) \in \{0,1\}^{k_F}} (-1)^{\sum_{i=1}^{k_F} \varepsilon_i} \\
&= \sum_{\{F \in \mathcal{G}(M_0) : k_F \neq 0\}} \left( \text{sgn}(\sigma_{M_0(F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0)}) \prod_{e \in F^{(\varepsilon_1, \dots, \varepsilon_{k_F})} \setminus M_0} a_e \right) (1-1)^{k_F} \\
&= 0.
\end{aligned}$$

Thus,

$$\begin{aligned}
\text{Pf}(A) &= \sum_{\{F \in \mathcal{G}(M_0) : k_F = 0\}} w_{M_0}(F \setminus M_0) \\
&= \sum_{\{F \in \mathcal{G}(M_0) : k_F = 0\}} \text{sgn}(\sigma_{M_0(F \setminus M_0)}) \prod_{e \in F \setminus M_0} a_e.
\end{aligned}$$

The set  $\{F \in \mathcal{G}(M_0) : k_F = 0\}$  consists of RCRSFs compatible with  $M_0$  containing no unicycles, and satisfying Condition (C). Observing that:

- the set of RCRSFs compatible with  $M_0$ , containing no unicycle is exactly the set of spanning forests of  $G^R$  compatible with  $M_0$  of Section 1.1 of the introduction,
- in the case of spanning forests, the partial reverse algorithm is exactly the trimming algorithm of Section 1.1,
- Condition (C) of Definition 1.2 and Condition (C) of Definition 3.1 are the same in the case of spanning forests,
- the permutation  $\sigma_{M_0(F \setminus M_0)}$  obtained from the algorithm is the permutation of Definition 1.1,

we deduce that  $\{F \in \mathcal{G}(M_0) : k_F = 0\} = \mathcal{F}(M_0)$ , and thus conclude the proof of Theorem 1.

EXAMPLE. Let us take the reference matching  $M_0$  which followed us throughout the paper, and consider the three possible perfect matchings  $M_1, M_2, M_3$  of the graph  $G$  given in Figure 2. Figure 9 shows the output of the full algorithm with initial superimposition  $M_0 \cup M_1$ . Since the superimpositions  $M_0 \cup M_2$  and  $M_0 \cup M_3$  contain doubled edges only, the output of the algorithm with these respective initial superimpositions, are the configurations themselves. By the Theorem 1, configurations  $M_2$  and  $M_{1,2,1}$  have opposite weights, and configurations  $M_3$  and  $M_{1,3,1}$  as well, so that their contributions cancel in the Pfaffian. As a consequence, signed weighted half-spanning trees counted by the Pfaffian of the matrix  $A$  are those of Figure 10 below.

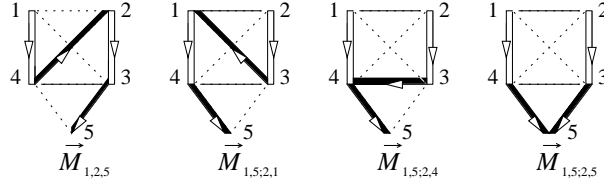


Figure 10: Black edges of the above configurations are half-spanning trees counted by the Pfaffian of the matrix  $A$ .

### 3.2 Proof of Corollary 3

Let us recall the setting:  $A^R$  is a skew-symmetric matrix of size  $(n+r) \times (n+r)$ , whose column sum is zero, with  $n$  even;  $A$  is the matrix obtained from  $A^R$  by removing the  $r$  last lines and columns;  $G^R$  and  $G$  are the graphs naturally constructed from  $A^R$  and  $A$  in the introduction.

Recall, see Section 2.1, that the signature of the permutation  $\sigma_M$  assigned to a perfect matching  $M$  counted by the Pfaffian of  $A$ , depends on the ordering of the two elements of pairs involved in the perfect matching, but not on the ordering of the pairs themselves. Choosing the signature of  $\sigma_M$  thus amounts to choosing an orientation of edges of the perfect matching  $M$ . The Pfaffian of  $A$  can thus be written as:

$$\text{Pf}(A) = \sum_{M \in \mathcal{M}} \text{sgn}(\sigma_M) \prod_{e \in M} a_e,$$

where the product is over coefficients corresponding to a choice of orientation of edges of  $M$ , specifying a choice of permutation  $\sigma_M$ .

Now, it is a known fact that the determinant of a skew-symmetric matrix is equal to the square of the Pfaffian:

$$\det(A) = \left( \sum_{M_0 \in \mathcal{M}} \text{sgn}(\sigma_{M_0}) \prod_{e \in M_0} a_e \right) \left( \sum_{M \in \mathcal{M}} \text{sgn}(\sigma_M) \prod_{e \in M} a_e \right).$$

As in Section 2.1, for every  $M_0 \in \mathcal{M}$ , we choose the permutation  $\sigma_M$  using the superimposition  $M_0 \cup M$ . Equation (1) thus yields:

$$\det(A) = \sum_{M_0 \in \mathcal{M}} \operatorname{sgn}(\sigma_{M_0}) \prod_{e \in M_0} a_e \left( \sum_{M \in \mathcal{M}} \operatorname{sgn}(\sigma_{M_0(M)}) (-1)^{|\mathcal{D}(M_0 \cup M)|} (-1)^{|\mathcal{C}(M_0 \cup M)|} \prod_{e \in M} a_e \right).$$

As a consequence of Theorem 1, this can be rewritten as:

$$\begin{aligned} \det(A) &= \sum_{M_0 \in \mathcal{M}} \operatorname{sgn}(\sigma_{M_0}) \prod_{e \in M_0} a_e \left( \sum_{F \in \mathcal{F}(M_0)} \operatorname{sgn}(\sigma_{M_0(F \setminus M_0)}) \prod_{e \in F \setminus M_0} a_e \right), \\ &= \sum_{M_0 \in \mathcal{M}} \sum_{F \in \mathcal{F}(M_0)} \left( \operatorname{sgn}(\sigma_{M_0}) \prod_{e \in M_0} a_e \right) \operatorname{sgn}(\sigma_{M_0(F \setminus M_0)}) \prod_{e \in F \setminus M_0} a_e. \end{aligned}$$

where  $\sigma_{M_0(F \setminus M_0)}$  is defined in Definition 1.1.

We have not yet chosen the permutation  $\sigma_{M_0}$  assigned to  $M_0$ , we do so now. For every  $F \in \mathcal{F}(M_0)$ , we chose the orientation of  $M_0$  to be the orientation of edges induced by the spanning forest  $F$ : this is precisely  $\sigma_{M_0(F \setminus M_0)}$ . Combining the product of coefficients  $a_e$  over oriented edges in  $M_0$  and in  $F \setminus M_0$  yields:

$$\begin{aligned} \det(A) &= \sum_{M_0 \in \mathcal{M}} \sum_{F \in \mathcal{F}(M_0)} \operatorname{sgn}(\sigma_{M_0(F \setminus M_0)})^2 \prod_{e \in F} a_e. \\ &= \sum_{M_0 \in \mathcal{M}} \sum_{F \in \mathcal{F}(M_0)} \prod_{e \in F} a_e, \end{aligned}$$

thus proving Corollary 3.

*Remark 16.*

1. We now give an intrinsic characterization of  $\cup_{M_0 \in \mathcal{M}} \mathcal{F}(M_0)$ , not using reference perfect matchings.

Consider the trimming algorithm of Section 1.1 applied to general spanning forests of  $G^R$  (not assuming that they are compatible with a reference perfect matching  $M_0$ ). Since the reference perfect matching is not used in the algorithm, everything works out in the same way, and the algorithm yields a sequence of paths  $\lambda_{\ell_1}^1, \dots, \lambda_{\ell_1}^N$ . This yields the following more general form of Definition 1.2.

*Definition 3.2.* A spanning forest  $F$  of  $G^R$  is said to satisfy *Condition (C)* if each of the paths  $\lambda_1^1, \dots, \lambda_1^N$  obtained from the trimming algorithm has even length. Let  $\mathcal{F}$  denote the set of spanning forests of  $G^R$  satisfying Condition (C).

*Lemma 17.*

$$\mathcal{F} = \bigcup_{M_0 \in \mathcal{M}} \mathcal{F}(M_0).$$

*Proof.* By definition, we have the following immediate inclusion:  $\cup_{M_0 \in \mathcal{M}} \mathcal{F}(M_0) \subset \mathcal{F}$ .

If  $M_0$  and  $M'_0$  are two distinct perfect matchings of  $G$ , then  $\mathcal{F}(M_0) \cap \mathcal{F}(M'_0) = \emptyset$ . Indeed suppose there exists a spanning forest  $F$  in the intersection. Then, it must be

compatible with  $M_0$  and  $M'_0$ , meaning that it contains all edges of  $M_0 \cup M'_0$ . Since  $M_0$  and  $M'_0$  are distinct, the superimposition  $M_0 \cup M'_0$  must contain a cycle, yielding a contradiction with the fact that  $F$  is a spanning forest.

Thus it remains to show that given a spanning forest  $F$  satisfying Condition (C) there exists a perfect matching  $M_0$  such that  $F \in \mathcal{F}(M_0)$ , meaning that  $F$  is compatible with  $M_0$  and satisfies Condition (C) of Definition 1.2. Let  $F$  be a spanning forest satisfying Condition (C), and let  $\lambda_{\ell_1^1}, \dots, \lambda_{\ell_1^N}$  the sequence of paths obtained from the trimming algorithm. For every  $i \in \{1, \dots, N\}$ , let  $M_0(\lambda_{\ell_1^i})$  consist of half of the edges of  $\lambda_{\ell_1^i}$  such that  $\lambda_{\ell_1^i}$  alternates between edges of  $M_0(\lambda_{\ell_1^i})$  and edges of  $\lambda_{\ell_1^i} \setminus M_0(\lambda_{\ell_1^i})$ , starting from an edge of  $M_0(\lambda_{\ell_1^i})$ . Let  $M_0 = \cup_{i=1}^N M_0(\lambda_{\ell_1^i})$ . Then  $F$  is compatible with  $M_0$  and satisfies Condition (C) of Definition 1.2. It remains to show that  $M_0$  is a perfect matching. The edge configuration consists of  $|V|/2$  edges, since by construction, it consists of half of the edges of a spanning forest. Moreover, since each of the paths  $\lambda_{\ell_1^1}, \dots, \lambda_{\ell_1^N}$  has even length, no vertex is incident to two edges of  $M_0$ , thus proving that  $M_0$  is a perfect matching.  $\square$

As a consequence, Corollary 3 can be rewritten in the more simple form:

*Corollary 18.*

$$\det(A) = \sum_{F \in \mathcal{F}} \prod_{e \in F} a_e.$$

2. Let  $\Xi$  be the set of cycle coverings of the graph  $G$  by cycles of even length: a typical element  $\xi \in \Xi$  is of the form  $\xi = (C_1, \dots, C_k)$  for some  $k$ . Then, since the matrix  $A$  is skew-symmetric, the determinant of  $A$  is equal to:

$$\begin{aligned} \det(A) &= \sum_{\xi=(C_1, \dots, C_k) \in \Xi} \prod_{\{i: |C_i| \geq 4\}} (-1) \left( \prod_{e \in \vec{C}_i} a_e + \prod_{e \in \overleftarrow{C}_i} a_e \right) \prod_{\{i: |C_i|=2\}} (-1) a_e a_{-e} \\ &= \sum_{\xi=(C_1, \dots, C_k) \in \Xi} \prod_{\{i: |C_i| \geq 4\}} (-2) \left( \prod_{e \in \vec{C}_i} a_e \right) \prod_{\{i: |C_i|=2\}} a_e^2. \end{aligned}$$

It is also possible to prove Corollary 18 directly, without passing through the Pfaffian, by applying the full algorithm to doubled edges of configurations counted by the determinant, and by taking into account all edges instead of half of them.

### 3.3 Line-bundle matrix-tree theorem for skew-symmetric matrices

In the whole of this section, we change notations slightly, and we let  $A$  be a skew-symmetric matrix of size  $n \times n$ , whose column sum is zero, with  $n$  even;  $G = (V, E)$  denotes the graph associated to the matrix  $A$ .

We now state a line-bundle version of the matrix-tree theorem for skew-symmetric matrices of Corollary 3, in the spirit of what is done for the Laplacian matrix in [For93], [Ken11], but first we need a few definitions.

A  $\mathbb{C}$ -bundle is a copy  $\mathbb{C}_v$  of  $\mathbb{C}$  associated to each vertex  $v \in V$ . The *total space* of the bundle is the direct sum  $W = \oplus_{v \in V} \mathbb{C}_v$ . A *connection*  $\Psi$  on  $W$  is the choice, for each

oriented edge  $(i, j)$  of  $G$  of linear isomorphism  $\psi_{i,j} : \mathbb{C}_i \rightarrow \mathbb{C}_j$ , with the property that  $\psi_{i,j} = \psi_{j,i}^{-1}$ ; that is, we associate to each oriented edge  $(i, j)$  a non-zero complex number  $\psi_{i,j}$  such that  $\psi_{i,j} = \psi_{j,i}^{-1}$ . We say that  $\psi_{i,j}$  is the *parallel transport* of the connection over the edge  $(i, j)$ . The *monodromy* of the connection around an oriented cycle  $\vec{C}$  is the complex number  $\omega_{\vec{C}} = \prod_{e \in \vec{C}} \psi_e$ .

We consider the matrix  $A^\psi$  constructed from the matrix  $A$  and the connection  $\psi$ :

$$(A^\psi)_{i,j} = a_{i,j}^\psi = a_{i,j} \psi_{i,j}.$$

A *cycle-rooted spanning forest* of  $G$ , also denoted *CRSF*, is a an oriented edge configuration spanning vertices of  $G$  such that each connected component is a tree rooted on a cycle. In all that follows, we assume that cycles have length  $\geq 3$ . Edges of branches of the trees are oriented towards the cycle, and the cycle is oriented in one of the two possible directions.

Consider the partial reverse algorithm of Section 3.1 applied to a general CRSF  $F$ . Since the reference perfect matching plays no role in this algorithm, everything works out in the same way, and the algorithm yields a sequence of paths  $\lambda_{\ell_1^1}, \dots, \lambda_{\ell_1^N}$ , whose union corresponds to branches of  $F$ .

**Definition 3.3.** A CRSF of  $G$  is said to satisfy *Condition (C)* if all of the paths  $\lambda_{\ell_1^1}, \dots, \lambda_{\ell_1^N}$  obtained from the partial reverse algorithm have even length. Let us denote by  $\mathcal{G}$  the set of CRSFs satisfying Condition (C). Then,

For a generic CRSF  $F$  of  $G$ , let us denote by  $(C_1, \dots, C_k)$  its cycles.

**Corollary 19.**

$$\begin{aligned} \det(A^\psi) = \sum_{F \in \mathcal{G}} \left( \prod_{\{e \in \text{branch}(F)\}} a_e \right) & \left( \prod_{\{i: |C_i| \text{ is odd}\}} \prod_{e \in \vec{C}_i} a_e [\omega_{\vec{C}_i} - \omega_{\vec{C}_i}^{-1}] \right) \\ & \cdot \left( \prod_{\{i: |C_i| \text{ is even}\}} \prod_{e \in \vec{C}_i} a_e [2 - \omega_{\vec{C}_i} - \omega_{\vec{C}_i}^{-1}] \right). \end{aligned}$$

*Proof.* We expand the determinant of  $A^\psi$  using cycle decompositions, as we have done for the determinant of  $A$  in Point 2 of Remark 16. Since the matrix  $A^\psi$  is not skew-symmetric, we cannot omit odd cycles, and we let  $\Xi$  be the set of cycle decompositions of the graph  $G$ , that is the set of coverings of the graph by disjoint cycles. A typical element of  $\Xi$  can be written as  $\xi = \{C_1, \dots, C_k\}$ , for some positive integer  $k$ . Then, the determinant of the matrix  $A^\psi$  is:

$$\det(A^\psi) = \sum_{\xi = (C_1, \dots, C_k) \in \Xi} \prod_{\{i: |C_i| \geq 3\}} (-1)^{|C_i|+1} \left( \prod_{e \in \vec{C}_i} a_e \psi_e + \prod_{e \in \overleftarrow{C}_i} a_e \psi_e \right) \prod_{\{i: |C_i|=2\}} \left( (-1) a_e \psi_e a_{-e} \psi_{-e} \right).$$

Using the skew-symmetry of the matrix  $A$  and the fact that  $\psi_{-e} = \psi_e^{-1}$  this yields:

$$\begin{aligned} \det(A^\psi) = \sum_{\xi = (C_1, \dots, C_k) \in \Xi} \prod_{\{i: |C_i| \text{ is odd}\}} & \left( \prod_{e \in \vec{C}_i} a_e [\omega_{\vec{C}_i} - \omega_{\vec{C}_i}^{-1}] \right) \\ & \cdot \prod_{\{i: |C_i| \text{ is even} \geq 4\}} (-1) \left( \prod_{e \in \vec{C}_i} a_e [\omega_{\vec{C}_i} + \omega_{\vec{C}_i}^{-1}] \right) \cdot \prod_{\{i: |C_i|=2\}} \left( \prod_{e \in \vec{C}_i} a_e^2 \right). \end{aligned}$$



Note that in a given covering there is always an even number of odd cycles, since otherwise there is no covering of the remaining graph by even cycles. We now fix a partial covering of the graph by odd cycles, and sum over coverings of the remaining graph by even cycles. Since the contribution of the parallel transport to doubled edges cancels out, and since the matrix  $A$  has columns summing to zero, we then ‘open’ doubled edges according to the full algorithm, using Remark 16. Everything works out in the same way, with the role of  $R$  played by odd cycles. In this case though, because of the parallel transport, the contribution of RCRSFs do not cancel, but looking at the proof of Theorem 1, we now precisely what those are. Summing over all partial coverings by odd cycles yields the result.  $\square$

*Remark 20.* Theorem 19 can then be specified in the case of bipartite graphs, in which case there are no odd cycles, in the case of planar graphs or of graphs embedded on the torus etc.

## References

- [Abd04] A. Abdesselam. The grassmann–berezin calculus and theorems of the matrix-tree type. *Advances in Applied Mathematics*, 33(1):51–70, 2004.
- [Cha82] S. Chaiken. A combinatorial proof of the all minors matrix tree theorem. *SIAM J. Algebraic Discrete Methods*, 3(3):319–329, 1982.
- [dT] B. de Tilière. Critical ising model and cycle rooted spanning forests: an explicit correspondence. *Arxiv:1012.4836. Accepted for publication. Comm. Math. Phys.*
- [Fis66] M. E. Fisher. On the dimer solution of planar ising models. *Journal of Mathematical Physics*, 7:1776–1781, October 1966.
- [For93] R. Forman. Determinants of Laplacians on graphs. *Topology*, 32(1):35–46, 1993.
- [HR04] S. Hirschman and V. Reiner. Note on the pfaffian matrix-tree theorem. *Graphs and Combinatorics*, 20(1):59–63, 2004.
- [Kas67] P. W. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, London, 1967.
- [Ken09] Richard Kenyon. Lectures on dimers. In *Statistical mechanics*, volume 16 of *IAS/Park City Math. Ser.*, pages 191–230. Amer. Math. Soc., Providence, RI, 2009.
- [Ken11] R. Kenyon. Spanning forests and the vector bundle laplacian. *The Annals of Probability*, 39(5):1983–2017, 2011.
- [Kir47] G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, 148:497–508, 1847.

- [KPW00] Richard W. Kenyon, James G. Propp, and David B. Wilson. Trees and matchings. *Electron. J. Combin.*, 7:Research Paper 25, 34 pp. (electronic), 2000.
- [KS04] Richard W. Kenyon and Scott Sheffield. Dimers, tilings and trees. *J. Combin. Theory Ser. B*, 92(2):295–317, 2004.
- [MV02] G. Masbaum and A. Vaintrob. A new matrix-tree theorem. *International Mathematics Research Notices*, 2002(27):1397–1426, 2002.
- [Tem72] H. N. V. Temperley. The enumeration of graphs on large periodic lattices. In *Combinatorics (Proc. Conf. Combinatorial Math., Math. Inst., Oxford, 1972)*, pages 285–294. Inst. Math. Appl., Southend, 1972.
- [TF61] HNV Temperley and M.E. Fisher. Dimer problem in statistical mechanics-an exact result. *Philosophical Magazine*, 6(68):1061–1063, 1961.